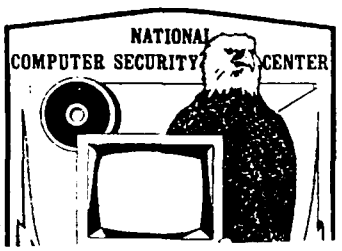
CSC-EPL-88/010



NATIONAL COMPUTER SECURITY CENTER

(2)

AD-A234 171

FINAL EVALUATION REPORT

OF

HEWLETT PACKARD

Computer [redacted] SYSTEMS DIVISION

MPE V/E

4 October 1988

Approved for Public Release:
Distribution Unlimited

Title change per phonecone with Brenda Anderson at National Computer Scty. Ctr., Ft. Meade, MD (301) 859-4452 on 4-12-91.

jk

A-1

FINAL EVALUATION REPORT

HEWLETT PACKARD COMMERCIAL SYSTEMS DIVISION

MPE V/E

NATIONAL
COMPUTER SECURITY CENTER

9800 Savage Road
Fort George G. Meade
Maryland 20755-6000

October 4, 1988

Library No. S231,332

This page intentionally left blank.

FOREWORD

This publication, the Final Evaluation Report Hewlett Packard Computer Systems Division, MPE V/E, is being issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center." The purpose of this report is to document the results of the formal evaluation of Hewlett Packard's MPE V/E operating system. The requirements stated in this report are taken from DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA dated December 1985.

Approved:

October 4, 1988

Eliot Sohmer
Chief, Office of Computer Security
Evaluations, Publications, and Support
National Computer Security Center

ACKNOWLEDGEMENTS

Team Members

Team members included the following individuals, who were provided by the Aerospace Corporation, El Segundo, CA

R. Leonard Brown, Ph.D.
James J. Donndelinger
Jeffrey R. Jones
Anne M. Wilson

Further Acknowledgements

Technical support was also provided by Deborah D. Downs, Ph.D.

Table of Contents

EXECUTIVE SUMMARY

The security protection provided by the Hewlett Packard Commercial Systems Division MPE V/E operating system, running on one of the HP3000 System Processing Units listed in Appendix A and configured in an appropriately secure manner as described in the Trusted Facilities Manual sections of the Security Management Guide[61], and running on an acceptable combination of hardware from the list in Appendix B has been examined by the National Computer Security Center (NCSC). The security features of MPE V/E were examined against the requirements specified by the DoD Trusted Computer System Evaluation Criteria[86] (the Criteria) dated 26 December, 1985 in order to establish a candidate rating.

The NCSC evaluation team has determined that the highest class at which MPE V/E appears to satisfy all the specified requirements of the Criteria is class C2.

A system that has been rated as being a C division system provides for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate. Such a system is expected to run in an environment of cooperating users processing data at the same level of sensitivity.

Systems in this class enforce discretionary access control (DAC) at the granularity of single users, making each user accountable for his actions through login procedures, auditing of security relevant events, and resource isolation.

The MPE V/E operating system has a number of features that enhance the security of the computing system. These include both hardware features such as virtual memory and a two state architecture, and software features such as Access Control Definitions (ACD) and individually restricted operating system commands. A summary list can be found at the beginning of the System Overview section (see page 6, "Security Features").

This page intentionally left blank.

INTRODUCTION

In April, 1987, the National Computer Security Center (NCSC) began a developmental product evaluation of MPE V/E, a product of Hewlett Packard Commercial Systems Division. It is intended that this report give evidence and analysis of the security features and assurances provided by the MPE V/E operating system. This report documents the evaluation team's understanding of the product's security design and appraises its functionality and integrity against the Criteria's C2 class security requirements. This evaluation applies to most HP3000 series System Processing Units (see Appendix A for the exact list) running MPE V/E release G.03.04 with patch number AV92, available to customers in October, 1988.

Evaluation Process Overview

The Department of Defense Computer Security Center was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In August 1985 the name of the organization was changed to the National Computer Security Center. In order to assist in assessing the degree of trust one could place in a given computer system, the DoD Trusted Computer System Evaluation Criteria was written. The Criteria establishes specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. The Criteria levels are arranged hierarchically into four major divisions of protection, each with certain security-relevant characteristics. These divisions are in turn subdivided into classes. To determine the division and class at which all requirements are met by a system, the system must be evaluated against the Criteria by an NCSC evaluation team.

The NCSC performs evaluations of computer products in varying stages of development from initial design to those that are commercially available. Product evaluations consist of a developmental phase and a formal phase. All evaluations begin with the developmental phase. The primary thrust of the developmental phase is an in-depth examination of a manufacturer's design for either a new trusted product or for security enhancements to an existing product. Since the developmental phase is based on design documentation and information supplied by the industry source, it involves no "hands on" use of the system. The developmental phase results in the production of an Initial Product Assessment Report (IPAR). The IPAR documents the evaluation team's understanding of the system based on the information presented by the vendor. Because the IPAR contains proprietary information, distribution is restricted to the vendor and the NCSC.

Products entering the formal phase must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal phase is an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the Criteria. The analysis performed during the formal phase requires "hands on" testing (i.e., functional testing and, if applicable, penetration testing). The formal

phase results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product successfully met all Criteria requirements in terms of both features and assurances. The final report and EPL entry are made public.

Document Organization

This report consists of four major sections and four appendices. Section 1 is an introduction. Section 2 provides an overview of the system hardware and software architecture. Section 3 provides a mapping between the requirements specified in the Criteria and the MPE V/E features that fulfill those requirements. Section 4 is a list of comments that the evaluation team thought were important to the evaluation process, but which did not fit into any of the other sections. Four appendices are included. The first two list the hardware and software components covered by this evaluation. Appendix C is a list of all reference material referred to in the report. The fourth contains the test descriptions

Conventions

To insure consistency with other NCSC reports and with Hewlett Packard Commercial Systems Division documentation, the conventions described in the following paragraphs were used in the preparation of this report.

Naming Conventions

The names of major sections of the operating system, referred to as separate products on the Master Installation Tape (MIT), are all in capital letters. For example, the MPE INSTALLER is all in capital letters. Individual programs, or relocatable program segments kept in the SL library file, are also all in capital letters. For example, SPOOK5, the spool handling utility program, is capitalized.

When referring to a specific name for an instruction, intrinsic, username, or file reference, all capital letters will be used. For example, MANAGER.SYS is a designated user recognized by the system, and PUB.SYS is the publicly accessible group of the SYS account. SL.PUB.SYS is a file within this group.

Small letters will be used to describe variable names, where actual values would be substituted in actual use. For example, username.acctname specifies that an actual username and account name must be substituted in the place where this phrase occurs in the description of a command. Periods are used as delimiters in the discussion of MPE V/E syntax, so the term SL.PUB.acctname refers to the segmented loader library of the PUB group of any account, but acctname must be replaced by an actual account name for actual use.

Source References

In general, when reference is made to an external source document, the first and most additional references to that document are accompanied by the notation [n], where the full bibliographic reference to that document is included as reference number "n" in Appendix C.

When all of the material in a paragraph or section has been drawn from a single external source document, this fact is often shown by the notation [n] at the very end of the final, or only, paragraph of the section, where "n" is the number of the document's full bibliographic reference in Appendix C.

This page intentionally left blank.

SYSTEM OVERVIEW

MPE V/E Background and History

Evolution of Hardware and Software

The HP3000 system can be described by the evolution of both its hardware and software. In 1974, the HP3000 CX Models 50, 100, 200 and 300 were introduced. These 16-bit stack architecture machines consisted of an SSI TTL CPU and up to 128Kb of core memory. The operating system which accompanied the CX models was MPE-C, the industry's first transaction oriented multi-programming operating system with an integrated data base management system.

The HP3000 Series II Models 5, 7, and 9 were introduced in 1976. Memory capacity began at 128Kb or 320Kb, depending on the model, with expansion possible up to 512Kb. 4K RAMS were used with standard error detection and correction ("fault control"). The CPU was MSI TTL. The system software introduced with the Series II, called MPE-II, provided for expanded memory support, system-to-system data communications support, and expanded code segment support.

In 1978, the HP3000 Series III was introduced. It provided memory capacity from 256Kb to 2Mb with a 16K RAM memory with standard fault control and a MSI TTL CPU. The MPE-III operating system included expanded memory support, interface bus (HP-IB) support, enhanced security, a help facility, private volumes, serial discs, labeled tapes, unified command language, and User Defined Commands (UDCs).

The HP3000 hardware evolution continued with the introduction of the Series 33 in 1978. It was the first application of Silicon-on-Sapphire technology to the HP3000. It supported the MPE-III operating system.

The HP3000 Series 44 was introduced in 1980. It provided between 1Mb and 4Mb of 16K RAM memory. The hardware also included a new second generation diagnostic computer, the Control and Maintenance Processor (CMP). With the Series 44, the MPE-IV operating system was introduced. Its new features included expanded memory support, new dispatcher/memory manager, look-ahead seeks, disc request priorities, interprocess communication, foreign disc facility, user logging, measurement interface, and telesupport.

The HP3000 Series 64 was introduced in 1981 with between 2Mb and 8Mb of 64K RAM memory, Writable Control Store, 32-bit data paths, dual 16-bit ALUs, and 32-bit memory transfers. A new diagnostic and control unit provided diagnostic capabilities. In 1983 the HP3000 Series 42, 48, 68 were introduced. These were disc caching versions of the Series 40, 44, and 64 respectively.

The MPE software evolution continued with the introduction of MPE-V in 1984 and 1985. It included expanded system table support for many more users, applications, and devices; disc caching support; expanded system directory; enhanced store/restore; rotational position sensing; and customer installability.

In 1985, the HP3000 hardware product family was broadened with the introduction of the HP3000 Series 37, an entry level system processor. In 1985, the HP3000 Series 58 was introduced with the addition of a 32Kb instruction cache. In 1986, the HP3000 Series 70 was introduced with 16Mb memory; the operating system was enhanced to support an expanded Segment Library. That same year, the Micro 3000 and Micro 3000XE were introduced to replace the Series 37 and Series 42, respectively.

Security Features

The MPE V/E operating system has the following security features:

- Virtual memory, with separate, variable length code and data segments.

- A file system based on a structure of accounts and groups, which provides default access protection between subjects (users) authorized to each account, and files associated with the various groups within the account.

- An Access Control Definition (ACD) mechanism that allows the creator of a file to define the access to the file to the granularity of a single user. If the ACD is left off a file, appropriate security defaults apply.

- ACD for devices, which may be applied by the System Manager as owner of all system devices.

- A hardware implemented privilege mechanism that uses two state protection which allows only privileged processes to run privileged code. These processes may be called by unprivileged users, but on exit from the privileged process, a return to user (non-privileged) state occurs.

- Spooled output files, with default restrictions such that only the user who submitted the file, or a privileged user, may manipulate it (cancel, copy, change output device).

- File based interprocess communication.

- Command execution based on individual user capabilities, as defined by the System Manager or the user's Account Manager.

- Control of submission of batch jobs.

- Separate administrator roles at the system and account level that allow access to the total system resources, and to subsets of the system resources, to be controlled easily. An administrative interface, consisting of MPE V/E commands, is provided for the System Manager and the individual Account Managers.

- Object reuse applied to all storage objects in the system.

Administrative Background

Because of the diversity of hardware bases on which MPE V/E will run, these systems are used in a wide variety of physical environments with various types of administrative setups. Thus, it is difficult to describe a typical physical installation. Naturally, for a secure computer system, the processing unit and console will be in a physically secure location, called a physical control zone, and terminal lines will be protected. However, the smaller systems, even when installed in a physical control zone, typically do not have an operator whose permanent job is to operate the machine. Typically, one or more users are trained in such operations, and most users have a clearance which allows them the necessary physical access to mount tapes, etc. Large machines, such as those used in a large timesharing environment, will have an operator who brings the system up, and performs services such as mounting tapes and serial discs.

Administratively, MPE V/E has several well-defined roles that are specified by user privileges. A user with System Manager privilege may create accounts. Accounts are used to allocate system resources and define to the system the users who may access those resources. The system Administrator may specify capabilities for each account. In a large installation, more than one such administrator may exist; in most installations only one is needed. For each account, there is an Account Manager who may add users into the account and specify capabilities to each user up to the maximum set of capabilities for that account. There is typically one Account Manager per account, and these personnel perform much of the everyday administrative work. It is also possible

for the System Manager to perform any task an Account Manager can perform, so after setting up an account the Account Manager can even be deleted, allowing the System Manager to perform all necessary administrative work. The Operations privilege allows a user to perform systems supervisor tasks, including installing new system software, tuning the system performance, and performing backups. In a small system several of the users might be given this privilege; in a large system a separate staff of system programmers would have it.

Hardware Architecture

Major Hardware Components

The MPE V/E runs on a wide range of hardware bases. For a complete list of the evaluated hardware components see page A-1, "Evaluated Hardware Components." Since all the machines listed in the configuration execute the same machine instruction set, the following is a generic discussion on the hardware architecture. The differences that exist in the hardware configuration range involve where the firmware that interprets the machine instruction set is stored. The Series 42, 42XP, 48, 52, 58, MICRO 3000 and MICRO 3000XE store the microcode in ROM. The top of the line machines, the Series 64, Series 68, and Series 70 load in the microcode from tape when booting up the system. This is referred to as writable control store. Microcode can be modified on the tape and reloaded into the machine during a reboot. Once loaded into the machine the microcode cannot be modified. The top of the line machines also include a CPU cache and dual ALUs. The cache is 8 Kbytes and data is transferred from cache to main memory and from main memory to cache in eight word blocks. Whether or not the cache has the desired word is transparent to the CPU, except in how long it takes to process the request. At any time there can be up to two requests to the cache pending, one for each ALU. In MPE V/E a dual ALU design is used to increase arithmetic processing power. This design allows the CPU to perform either two 16-bit or a single 32-bit operation in one CPU cycle.

The hardware elements of the computer system are in independent modules attached to a Central System Bus. The I/O system includes two major hardware entities: Intermodule Bus (IMB) I/O Adapters, and Intermodule Bus I/O Channels. This allows modules to run independently at their own speed. This structure also allows new equipment to be added without going through a major hardware reconfiguration. The separate I/O busses allow many I/O operations to be handled concurrently with CPU and Main Memory operations. DMA can be accomplished through the use of a General I/O Channel (GIC). The GIC is an I/O channel controller which talks to the Intermodule Busses and has HP-Interface Bus (HPIB) devices attached to it. For more detailed information see page 20, "I/O Manager."

MPE V/E runs on a stack machine which uses segmented virtual memory. Variable length code and data segments are maintained by MPE V/E; separate tables exist for all data segments being referenced by currently loaded processes, and for all code segments that are referenced by currently loaded programs. A program may be run by more than one process at any given time; each process has its own data segments, but shares the code segments of the program it is running. Code segments are non-modifiable and re-entrant. Since the code is non-modifiable it is overlayed in main memory and does not get written back to disk. Data segments are dynamic and are swapped out to disk when necessary. See page 12, "Address Translation" for more information on code and data segments.

MPE V/E operates on a two state machine, user mode and privileged mode. Hardware maintains

the separation of operating modes by using a bit in a process' status word. The basic addressing mode is word addressing, 16 bits, with provisions for instructions to load and store bytes and double words. The CPU provides 72 specific purpose registers, 14 of which are accessible to a user process. Registers are 16 bits in length and most of the user accessible registers are used for defining segment limits and operating elements within a segment. The contents of a register is changed by the microcode as it interprets the machine language instructions. A user process cannot directly modify any register value unless it uses the appropriate instruction. A brief description of each of the 14 user accessible registers follows.

PB	Program base; lowest or base address of the code segment
PL	Program limit; highest or limit address of the code segment.
P	Program counter; location of currently executing instruction in the code segment
DL	User data limit; lower bound of the stack user area.
DB	Base reference point for data; marks the beginning of the storage for variables and parameters.
X	Index register.
STA	Status register.
Q	Base of the current dynamic stack area.
Z	Current stack limit; upper bound of the stack user area.
RA	First element of stack (i.e. top of stack).
RB	Second element on the stack.
RC	Third element on the stack.
RD	Fourth element on the stack.
S	Top of stack pointer.

Figure 2.2.1 illustrates the relations of most of these registers to a simple user process address space.

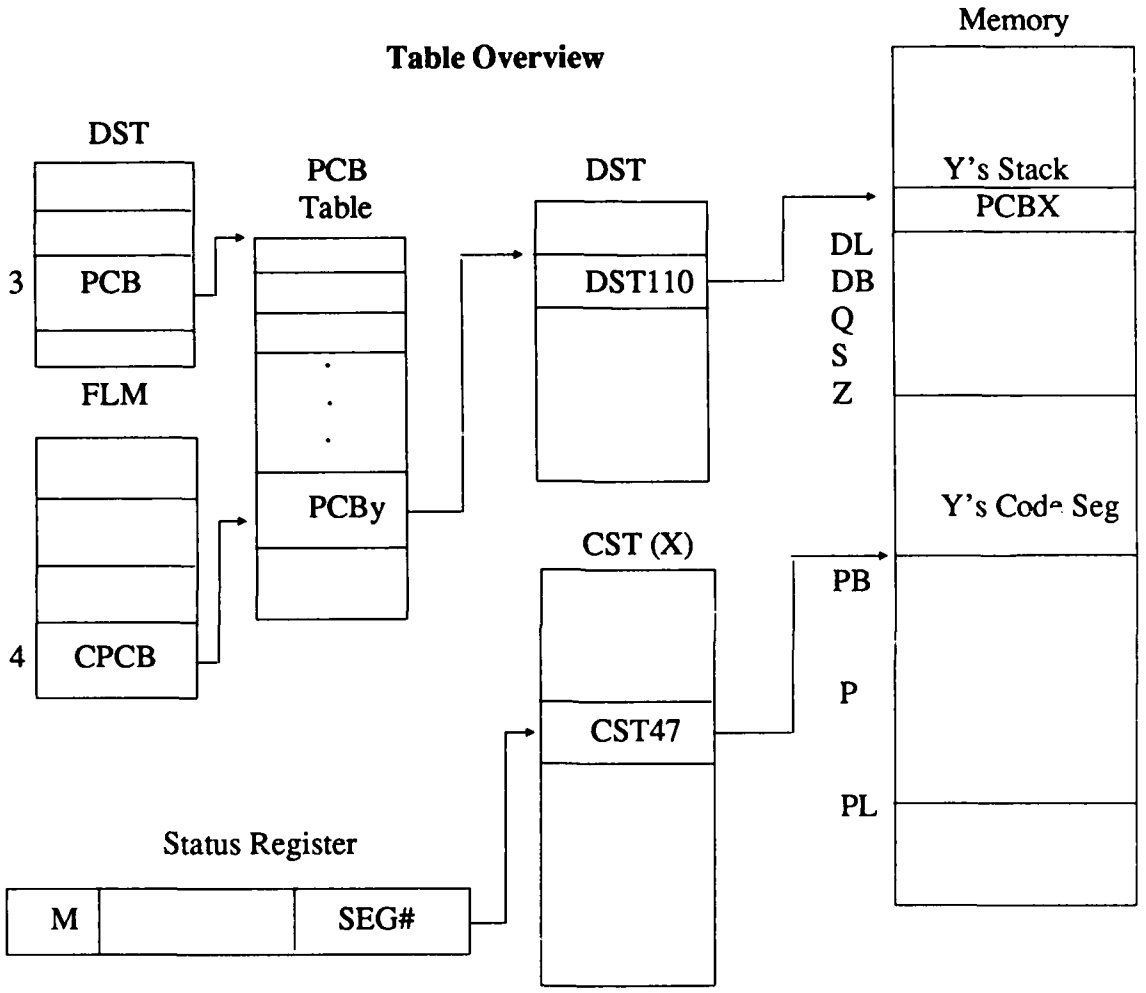


Figure 2.2.1 Code Segment and Stack Segment of Simple Process

Processor Address Space

In MPE V/E, privileged mode programs have the ability to execute privileged instructions, may access the 58 registers not accessible in user mode, and may make program and data references without bounds checking. This allows privileged programs to reference and manipulate any absolute address location. For example, privileged instructions allow moving a specified number of words in or out of any specified data segment, see page 12, "Address Translation" for more information on data segments. A privileged process may also move, load or store from or into absolute addresses with no limitations.

Process Address Space

A process in MPE V/E consists of a unique stack, possibly one or more extra data segments (XDS), one or more code segments, and an entry in the Process Control Block (PCB) table. See Figure 2.2.1. An XDS is used if a process needs more memory for data than its stack allows or if data is to be shared between processes in a session. See page 12, "Address Translation" for more information on segments. The PCB entry contains the minimum information that the dispatcher needs in order to decide which process should be given the CPU next. For example, the PCB entry informs the dispatcher if the process is ready to run and only needs the CPU allocated to it, or if the process is blocked, waiting on I/O or some other resource. Once a process is selected for execution, the PCB entry will tell the dispatcher which data segment table (DST) entry describes the stack.

In MPE V/E, the stack (user data segment) is the principle data structure for each process and has a maximum size of 64K bytes and a minimum size of 8 bytes. When a process is executing, its stack, the code segment currently being referenced, and the extra data segment (XDS) being referenced, if any, must be in main memory. Five hardware registers define the bounds of a process: DB, DL, Q, S, and Z. The current code domain of a process is defined by three hardware address registers: PB, PL, and P.

The stack of each process contains the bulk of the process information in an area referred to as the Process Control Block Extension (PCBX) The PCBX is an extension of the PCB entry and is located in the process' stack starting at location 0 and extending to but not including DL. This area of the stack contains information for managing extra data segments, as well as other information. See page 33, "Process Control" for more information on the PCBX area of the stack.

Address Translation

Since MPE V/E is a stack machine, it basically uses a base and bounds translation with some segmentation. When referencing data in the stack, user programs make references to the Top of Stack (TOS) implicitly or can explicitly make references relative to DB, Q, or S (TOS). A process

may also make references in its code segment relative to P. Memory address instructions use bits 6 through 15 for mode and displacement and addressing can be + or - relative to P or Q, but only + relative to DB and - relative to S. The relative addressing displacement ranges for the various modes are shown below.

P relative	+/- 255 locations	
DB relative	+ 255 locations	
Q relative	+127 locations	-63 locations
S relative	-63 locations	

The CPU routinely checks all address references and TOS movements to ensure that such operations remain within legal bounds. Sufficient checks are made for all machine instructions to ensure that a non-privileged user cannot adversely affect other users or the operating system. If any of the bounds checks fail during non-privileged user mode, there will be a bounds violation interrupt. There are five types of checks made: program transfer limit, program reference limits, data reference limits, stack overflow limit, and stack underflow limit.

Program Transfer Limit.

Program control cannot be passed to any location beyond the limits defined by the contents of the PB and PL registers. For indirect branches, both the indirect and direct references must be within limits.

Program Reference Limits.

Some of the memory address instructions, all loop control instructions, and some move instructions are capable of addressing locations in the code segment. During privileged mode, these references can be made as desired. During non-privileged user mode these references, both direct and indirect, must be within the limits defined by PB and PL.

Data Reference Limits.

During privileged mode, data references are not subject to bounds checking. During non-privileged user mode these references, both direct and indirect, must be within the user's area defined by DL and S.

Stack Overflow Limit.

Stack overflow is defined as moving the S-pointer beyond the stack limit. A stack overflow causes an interrupt. The interrupt handler may, at the discretion of the operating system, extend the stack limit.

Stack Underflow Limit.

Stack underflow is defined as moving the S-pointer below DB. During privileged mode, stack underflow is not subject to checking. During non-privileged user mode, stack underflow causes an interrupt.

Extra data segments (XDS) can be allocated to a user by executing the intrinsic GETDSEG. The maximum and minimum size of a XDS is the same as the stack, 64K bytes and 8 bytes. When a XDS is allocated to a user, the operating system assigns a token to an index number in the data segment table (DST) and places the pair in the process' PCBX area, and returns the token representing that XDS to the user. See Figure 2.2.2. In user mode, the only way to access a XDS is through two intrinisics, DMOVIN and DMOVOUT. These intrinsics allow the user to move data between the stack and XDS. When making references to a XDS the user specifies the token and the operating system fetches the corresponding index into the DST from the PCBX area of the process' stack. When referencing data within a XDS, all references are relative to DB for that segment. In the privileged mode the system moves the DB register from the stack to the XDS or vice versa using a DB exchange instruction. This usage is referred to as "split stack mode" operation.

Every data segment, whether a process' stack, an XDS, or a system data segment, is tracked by the DST. The DST is addressed by microcode through the third word in fixed low memory. The index to the DST for a process' stack is located in the PCB entry, while the index for a XDS is located in a process' PCBX. The DST entry contains information on the segment location, segment length, and a flag that indicates whether the segment is present or absent from main memory.

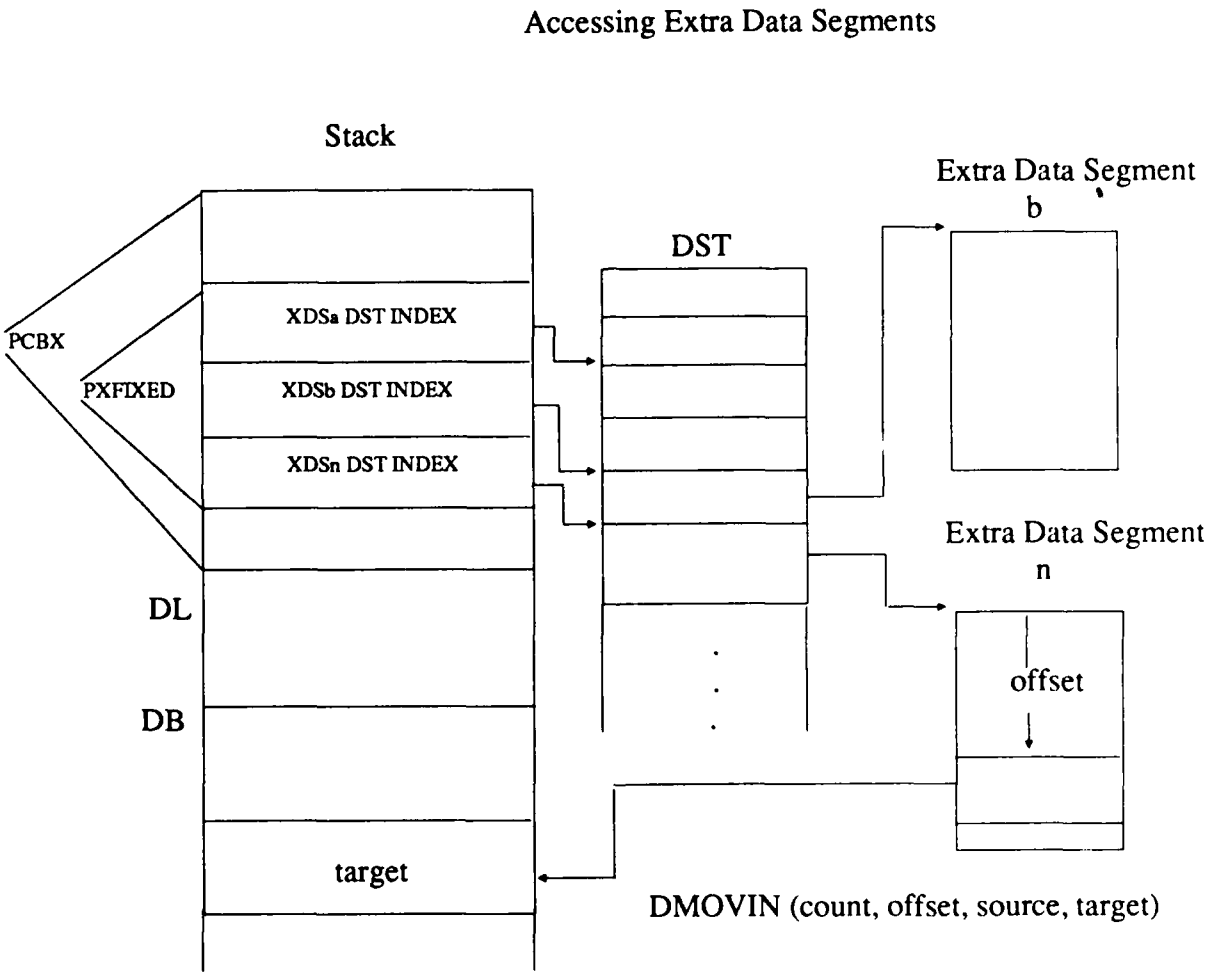


Figure 2.2.2 Accessing XDS

Code segments are divided into segment library (SL) segments and program segments. The maximum size of a code segment is 32Kbytes and the minimum size is 8 bytes. SL segments are tracked by a code segment table (CST) and program segments are tracked by a code segment table extension (CSTX). A SL segment is a special type of code segment created by SEGMENTER. It has the property that its procedures may only call other procedures within the same or within other SL segments. The main use of a SL segment is to contain system procedures that are invoked by calls to intrinsics. For example, a single SL segment may contain all the code for the file system intrinsics such as FOPEN, FREAD, FWRITE and FCLOSE. A program may also have its own SL segments to support special needs. See Figure 2.2.3 for an illustration of the format of the CST and CSTX tables. The CST looks the same to all users, while the CSTX appears unique to each process that is running each program. The PCB entry for each process contains a word that is an offset in the CST Block Table (CSTBT) and also contains how many entries or segments a program has in the CSTX. The CSTBT entry contains an offset into the CSTX which points to a program's block in the CSTX. Each process that is running that program has a pointer to the same CSTBT entry. All references in the CSTX are relative to the CSTBT pointer for each process. Microcode checks to assure a process does not cross its boundary. The CST is addressed by microcode through the first word in fixed low me.nory. The CSTX is addressed through location 5 of the DST. The CST and CSTX entries are identical in format and contain information on the segment location, type (privileged, system, or user), segment length, and a flag that indicates whether or not the segment is present in main memory.

Each code segment has a Segment Transfer Table (STT) contained within itself. See Figure 2.2.3 for an illustration of the STT. This table is used when making a procedure call (PCAL). The STT contains an entry for each PCAL in that code segment. When making a PCAL, an index in the STT is imbedded in the instruction. The STT is broken into two sections: local labels and external labels. If the desired PCAL is in the same segment (local), the address of the call is located in the STT entry. If the PCAL is located in another segment, the entry in the STT contains the segment number and the index into that segment's STT. Also associated with the STT entry is a bit which determines whether or not a PCAL is callable by a user mode process.

Accessing User Code Segments

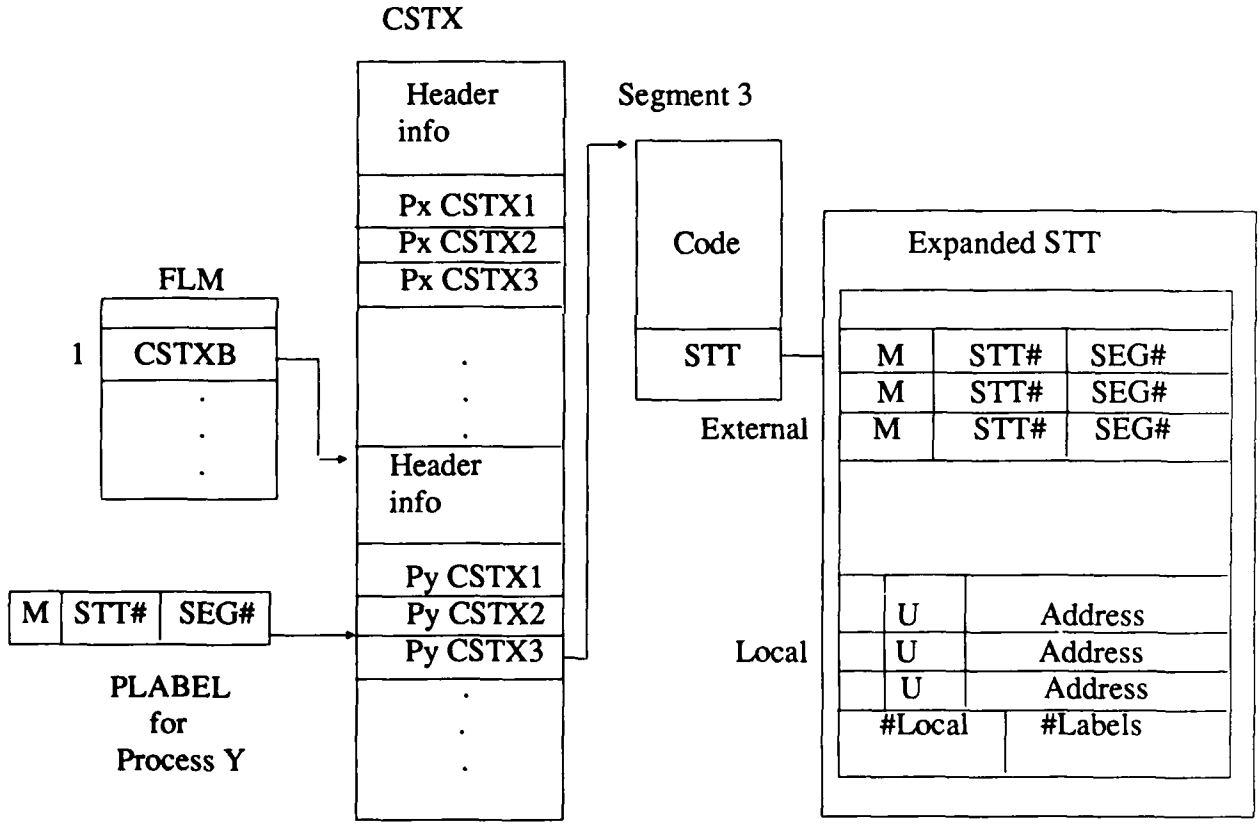


Figure 2.2.3. CSTX and STT Tables

When a process references a procedure within a segment library (SL), the microcode determines from the STT whether it is a physical SL segment, which are reserved for system SL's, or if it is a logical (user) SL. If it is a logical SL reference, the microcode makes a logical to physical transformation. A logical segment transform table (LSTT) is used to make the mapping. See Figure 2.2.4. All processes that access user SL segments have an associated LSTT and it is pointed to by the process' PCB entry. User program segments are assigned segment numbers in the LSTT starting from 1 and going to n where n is the number of user segments in the program. This is done so user segments do not have to be mapped to a physical location. The remaining entries in the LSTT are similar to an external STT entry. The entry contains a desired segment number and a offset into the STT for that segment. A LSTT can be shared by multiple processes if they are running the same program.

Domain Separation

In MPE V/E there are two states of operation: privileged and user. This is controlled by the Process Status word. There is a status word for each code segment in the system. At all times, the status word associated with the segment currently executing indicates the machine status following the execution of the most recent instruction in that segment. The status for the currently executing segment is resident in the status (STA) register and is constantly being updated as each instruction is executed. For segments that are not current (suspended by an interrupt or procedure call), the status word exists in a stack marker in the user's data stack. While on the user's stack this bit can be modified by the user but microcode checks prevent the user from receiving privileged mode. When a process calls a privileged segment its stack marker is pushed on the stack. A call is successful if the caller is privileged or if the uncallable bit, which is associated with each privileged segment, is not set. Upon exiting, the user's stack marker is popped off the stack. The microcode checks the calling code segment's mode and allows the user to exit in privileged mode only if the calling code segment was in privileged mode. If a process modifies its stack marker from user mode to privileged mode, the microcode detects the action and issues a "Privilege Violation" and aborts the process. The only way a process can exit with privileged mode is if it was privileged when making the call. Bit 0 in the status register is used to indicate whether the current segment is running in privileged mode (bit 0 = 1) or user mode. The state of this bit cannot be changed by machine instructions while resident in the STA register except in privileged mode. Bit 1 is

Code Segment Transfer

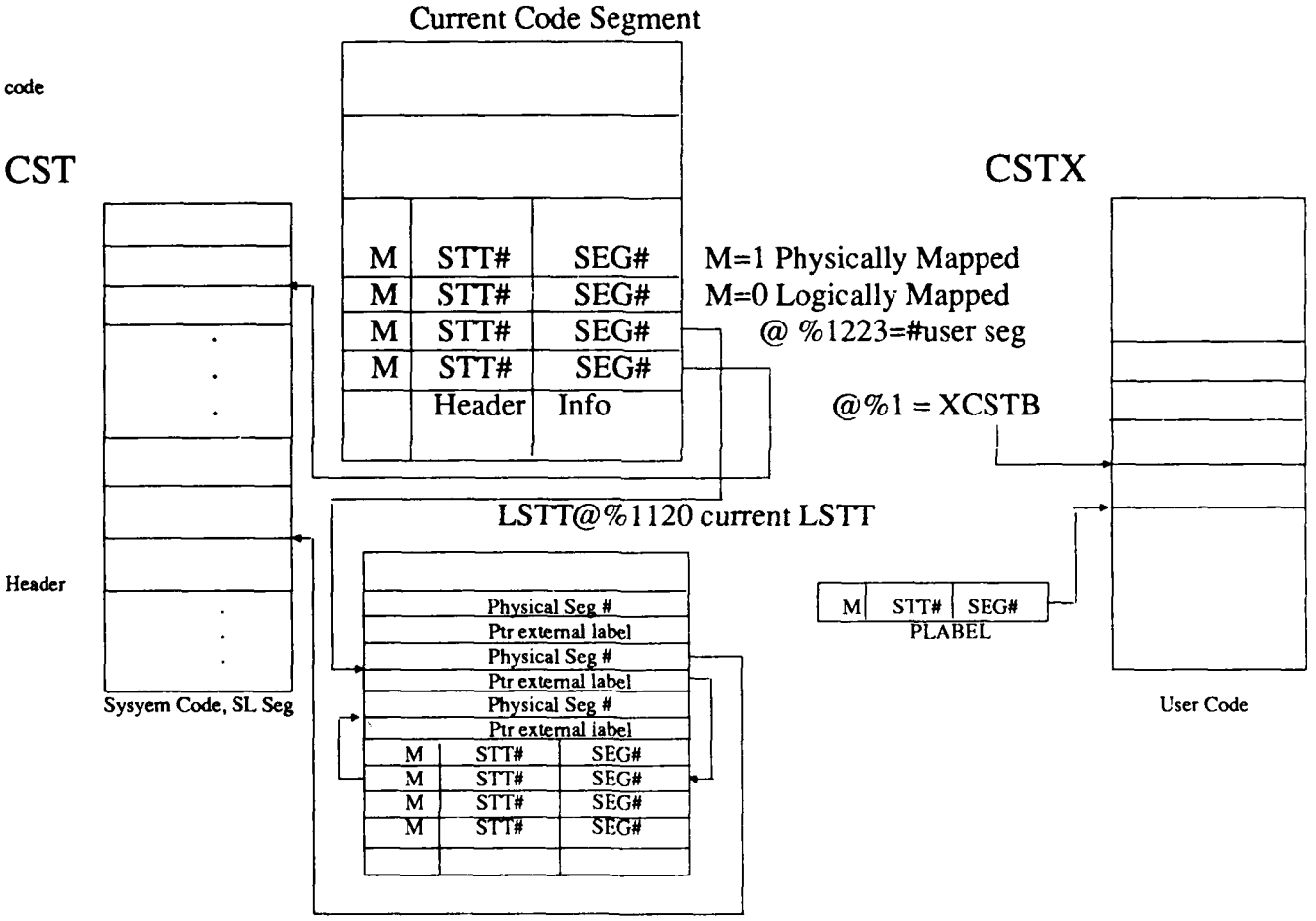


Figure 2.2.4 Code Segment Transfer

used to enable or disable external interrupts. The state of this bit can only be changed in privileged mode. Bit 2 is used to enable or disable user traps. The state of this bit can be changed in any mode while current or non-current with a SETR instruction. The remaining bits are used for overflow and condition codes.

Interrupts

There are two basic types of interrupts in MPE V/E: internal and external. Internal interrupts are a result of the CPU and its microcode detecting some unusual or abnormal condition. Some examples of internal interrupts are stack overflows, absence traps, integer overflows, and bounds violations. In certain cases, such as a stack overflow, the interrupt control stack (ICS) is used by the procedure that processes the interrupt, but in most cases, internal interrupts are processed on the user's stack. This allows an unprivileged user to write his own interrupt handler for such events as divide by zero and integer overflow. Internal interrupts that are handled on the process stack are termed "traps" while those that are handled on the ICS are called "interrupts." External interrupts are used to indicate a change in status of an I/O device and are always processed on the ICS. The type and interrupt number of these interrupts depends on which I/O hardware is installed on the system, and how the hardware is configured.

The microcode interrupt handler determines what type of internal interrupt has occurred and must invoke the software interrupt handling procedure to take the appropriate action. This is accomplished by an internal branch to the appropriate procedure call. The microcode knows which STT number, or procedure, to call in which code segment, since the interrupt handling procedures are located in code segment 1, referred to as ININ, and the corresponding STT is always constructed to have the same order expected by the microcode. The microcode determines which interrupt handler to branch to for external interrupts by using the PLABEL in the interrupting device's device reference table (DRT) entry.

The interrupt control stack (ICS) is a memory-resident stack used by all external and some internal interrupts. This data structure is considered the "home stack" for the dispatcher. (See page 34, "Process Switching.") If an ICS type interrupt occurs, a special 6-word marker is pushed onto the user's stack. Control is then transferred to the ICS with the parameter or DRT number located at Q+3 on the ICS. The IXIT instruction is executed at the end of every interrupt handler. This returns control to the interrupted process.

I/O Manager

The MPE V/E I/O System consists of two major hardware entities: Intermodule Bus I/O Adapters and Intermodule Bus I/O Channels. See Figures 2.2.5 and 2.2.6 for an illustration of the hardware organization. The Intermodule Bus Adapter includes the I/O Buffer, Intermodule Bus Interface, and Common Bus Interface. The I/O Channels are the General I/O Channel (GIC) and the Advanced

Terminal Processor (ATP). The Hewlett-Packard Interface Bus (HP-IB) is the bus that is used to connect peripherals to the GIC.

The Intermodule Bus Adapter (IMB IOA) converts synchronous, block-sized transfers on the Central System Bus to asynchronous, word-sized IMB-compatible transfers, and vice versa. There can be a maximum of eight IMB IOAs and a minimum of one. The IMB IOA also controls DMA between main memory and the I/O channels, converts IMB Channel Service Requests and Interrupt Requests into messages to the CPU, and interprets commands from the CPU.

The GIC is the primary channel for communication between the CPU and the I/O devices other than terminals. Each GIC communicates via the Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts, since I/O is achieved with channel programs. Software is responsible for setting up a channel program, but the execution is performed by the CPU's channel microcode. The device driver contains the code that builds a channel program. A skeleton already exists and the device driver fills in the control signals and provides the addresses the device is to access. The driver receives this information from the operating system. A user is unable to create, modify or call a channel program directly.

Each channel on the IMB may support up to eight devices, and each of these devices may have a unique, dedicated channel program controlling its operation. These programs consist of specialized channel instructions which are completely unrelated to the CPU instruction set. All eight programs on each channel may be "running" simultaneously, although only one may be executing at a time.

The GIC contains DMA hardware which allows large records of data to be transferred at the maximum speed of the HP-IB. The channel microcode enables the device and then initiates the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the IMB and allows the GIC to perform its function through DMA operation. During this time the GIC becomes master of the IMB and the IOA and controls traffic flow.

The ATP is an intelligent interface between terminals and the CPU, and it operates similar to the GIC. An ATP is composed of a System Interface Board (SIB) and from one to eight Port Controllers. The SIB provides a hardware interface to the IMB and, under microprocessor control, performs byte packing and unpacking and controls DMA of user data. Port Controllers provide the hardware interface for terminal devices. With control logic dedicated to each terminal port, the Port Controller handles all handshaking between the system and the connected devices. The ATP allows terminals to transmit and receive data on either a character-by-character basis or a block-at-a-time basis. For both types of operations, the ATP transfers data directly to and from memory.

The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate I/O commands from the GIC to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the GIC. The device controller also generates interrupts when required by some device condition or by channel command.

Device controllers are identified by a logical device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software and contains, among other things, a pointer to the start of the SIO program for each device controller. Each device controller connects to a GIC. Certain device controllers may control several logical devices. In such cases, each logical device attached to the controller is addressed separately using a unit number assigned when the device is installed.

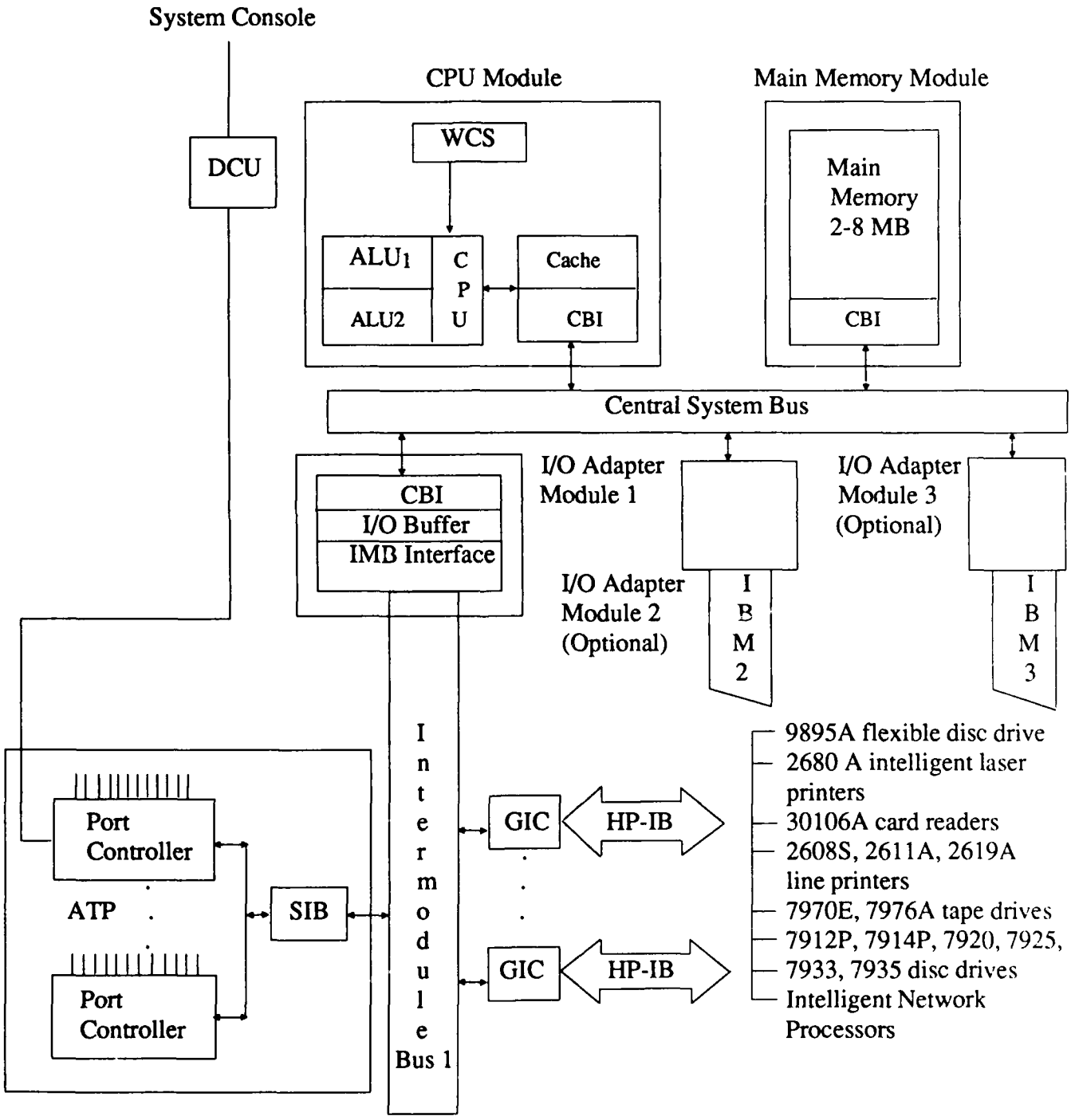


Figure 2.2.5 Hardware Organization for Series 64, 68 and 70.

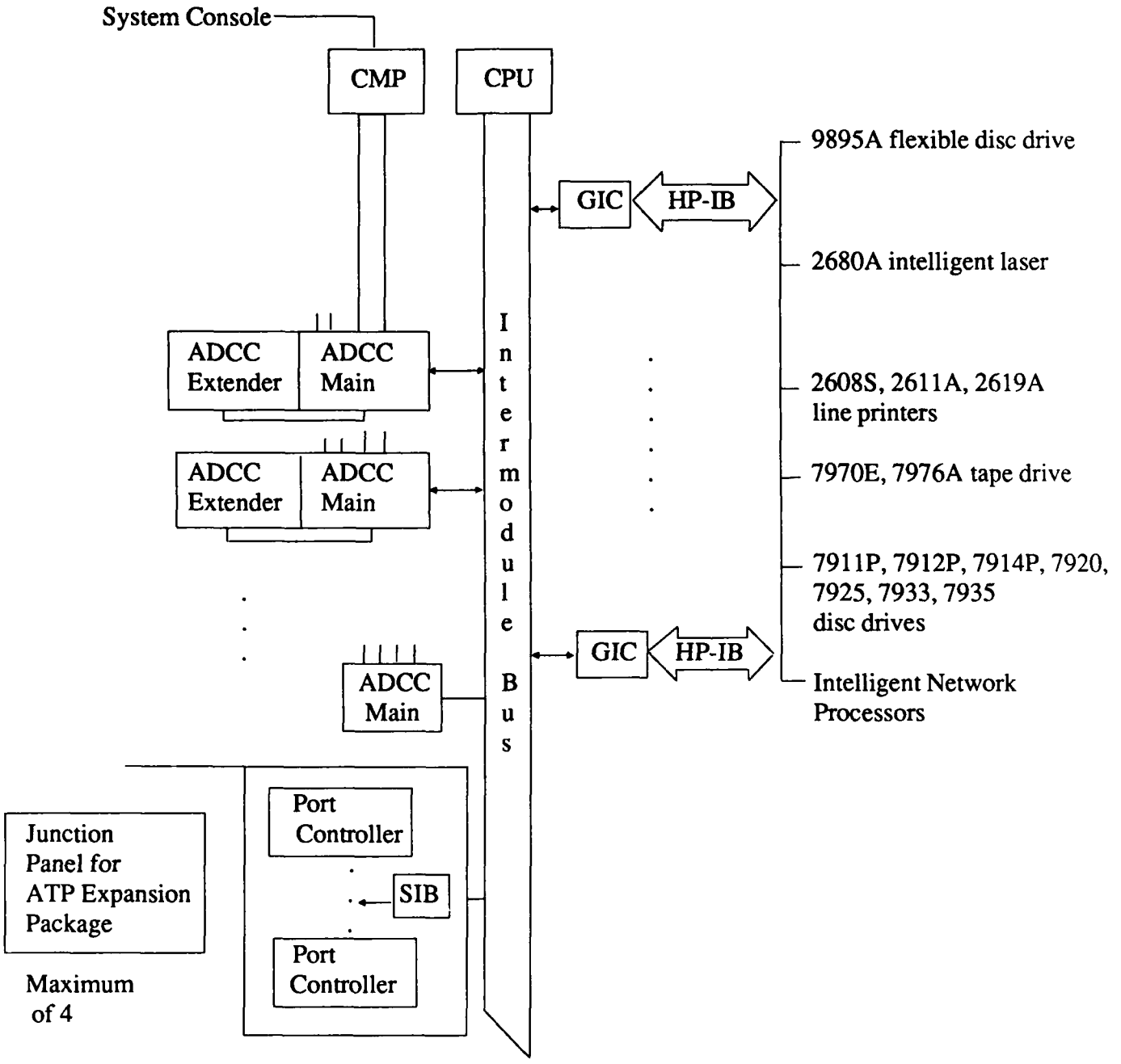


Figure 2.2.6 Hardware Organization for Series 42, 42XP, 48, 52, 58, MICRO 3000 and MICRO 3000XE.

Software Architecture

MultiProgramming Executive

The MultiProgramming Executive MPE V/E is a general purpose, disc-based operating system that supervises all processing and maintains all user interfaces on the HP 3000 computer system. MPE V/E dynamically allocates system resources such as main memory, the central processor, and peripheral devices to each process as needed. It also coordinates all user interaction within the system and provides a command interface in the form of a command language interpreter. It maintains records of all resources used. Through multiprogramming, MPE V/E can execute several different jobs and sessions at the same time.

Typical of all modern operating systems, MPE V/E supports interactive as well as batch processing. The version of MPE V/E that was evaluated includes protection of disc files and access to individual devices by means of Access Control Definitions (ACD), which are similar to the access control lists mentioned in the Criteria[86], but all system versions have an underlying system of protection based on matching the capabilities of a user with each file's specified capability requirements. This basic protection mechanism is used to maintain the security of the operating system components themselves.

MPE is designed around a structure of accounts, which are the basic resource unit of the system, and groups, which provide the local domains of disc files accessible by individual users. This system of accounts is used to distinguish user resources and files for both security and accounting purposes. Important system features include virtual memory utilizing separate, variable length code and data segments, disc caching, file based interprocess communication, and User Defined Commands (UDC) which can be processed by the Command Interpreter.

The MPE V/E File and Account System

An understanding of the organization of the Trusted Computing Base requires an understanding of the file and account system. The account structure consists of four components: accounts, groups, users, and files. The major component of the structure is the account, which is the identifiable unit to which system resources are allocated. Except for the SYS account which exists when the system is first brought up, each account must be created by a user with the System Manager capability. The System Manager then typically creates a user who has Account Manager capability with that new account. The Account Manager then creates each user by assigning a unique login name. Each individual user ID is identified uniquely with the account in which it was created. Each user is assigned certain capabilities when created; these may only be a subset of the capabilities of the account. The user MANAGER.SYS has capabilities built into the MPE V/E structure.

The SYS account has special capabilities that make its PUB group an appropriate home for both the TCB software and for other system software. For example, the group PUB.SYS was created so that any executable program stored within it may run in Privilege Mode, provided it was compiled with Privilege Mode, and PREPed with Privilege Mode. This group contains all the utilities of the operating system, including both those that run continuously, such as the SCHEDULER, and those that may be invoked, such as the INSTALLER. There is a file, SL.PUB.SYS, which contains loadable intrinsics (standard system procedures) which can be loaded with a user program. SL stands for segmented library, since the procedures that carry out the functions of the intrinsics calls are stored, one or more procedures to a segment, in loadable, variable length code segments. When a program is RUN, the user can specify the parameter

LIB=[G,P,S]

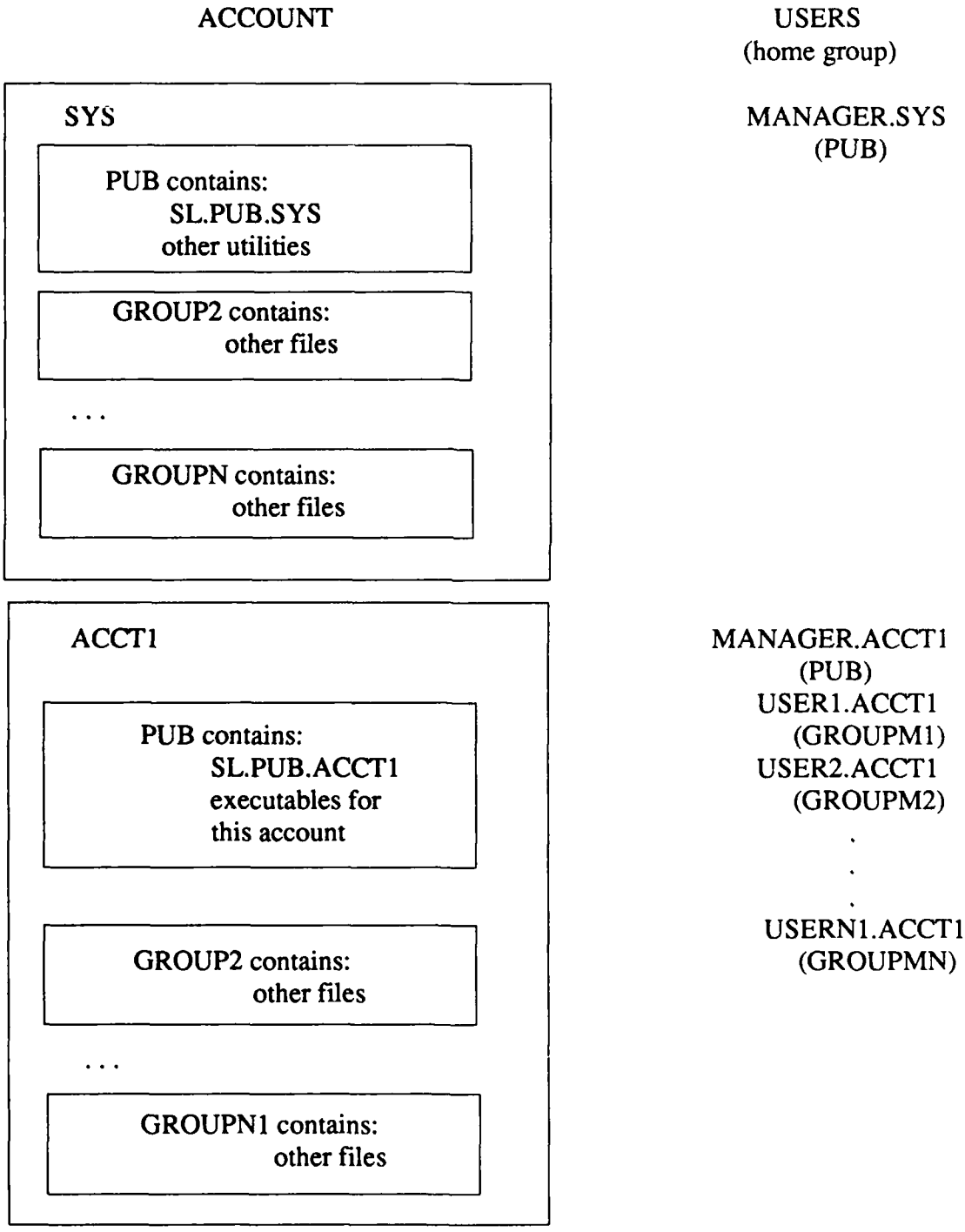
to determine the order in which libraries will be searched to resolve external references. A parameter value of S means that only SL.PUB.SYS will be searched. A value of P means that SL.PUB.acctname will be searched within the user's account before the system segmented library is searched. The value G means that SL.groupname.acctname is searched first for the group in which the user is logged, then SL.PUB.acctname, then finally the system segmented library.

Each account has a unique file domain which supplies users within that account certain default access rights to these files. The account is further divided into groups to partition both the account's accumulated resources and to divide the file domain into separate subdomains. Within every account, there exists a group PUB.acct to which all account users have some kind of access. This access is predetermined at account creation time and only changeable by a user with Account Manager capability. Within that group, there may be a file SL.PUB.acct which contains the Segmented Library of code segments which have already been processed by the SEGMENTER program and are ready to be loaded as separate virtual memory segments by the MPE V/E load utility. Other files that may be contained within the groups belonging to an account include data files, source language files, and object files output by the various source language compilers.

The file system procedures in MPE V/E control access to all files, including system files used by the operating system itself. There are three different types of files handled by MPE V/E. A file can be privileged sequential, unprivileged sequential, or keyed sequential (KSAM). A privileged sequential file has the high order bit set to 1 in its file identifier in the Directory (this is where system information such as account, user and file identifiers is stored, and is always on logical device 1), and only a process running in privileged mode can access such a file. An unprivileged file has its high order bit set to 0 in its file identifier. MPE V/E still uses the full access control decision mechanism to determine if a user process will be given access to a non-privileged file, but will not also require that the process be running privileged to gain access.

When MPE V/E handles access to Keyed Sequential Access Method (KSAM) files, it requires to use of special intrinsics (system procedures) which are part of the file system, but which are not required when accessing sequential files. KSAM files are stored in such a way that one can access data within a disc file directly, based on the ASCII value of a key associated with the data within the file.

Figure 2.3.1 illustrates the general structure of an HP 3000 system running MPE V/E.


ACCOUNT
USERS
(home group)
SYS
PUB contains:
SL.PUB.SYS
other utilities
GROUP2 contains:
other files
. . .
GROUPN contains:
other files
MANAGER.SYS
(PUB)
ACCT1
PUB contains:
SL.PUB.ACCT1
executables for
this account
GROUP2 contains:
other files
. . .
GROUPN1 contains:
other files
MANAGER.ACCT1
(PUB)
USER1.ACCT1
(GROUPM1)
USER2.ACCT1
(GROUPM2)
USERN1.ACCT1
(GROUPMN)

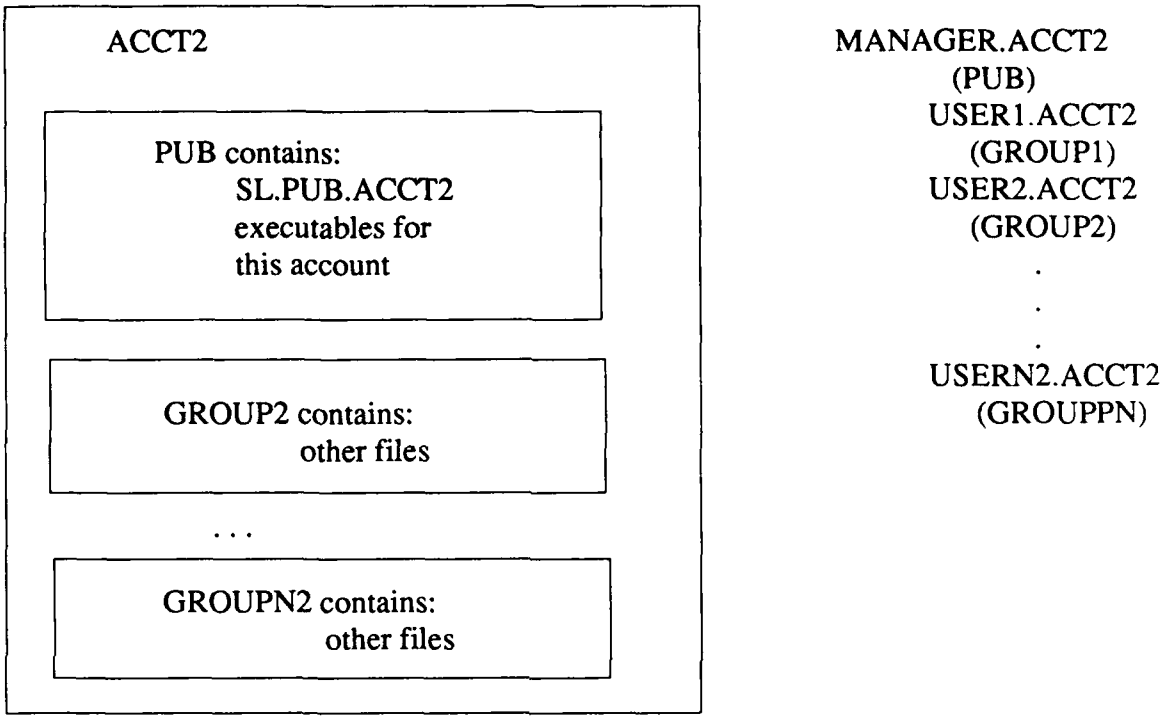


Figure 2.3.1 - Diagram of Group and Account Structure

TCB Components

Hewlett Packard refers to its MPE V/E operating system as its Fundamental Operating System (FOS), which includes not only the kernel MPE operating system functions and intrinsics, but also various utilities that are required to maintain the integrity and efficiency of the computer system. For administrative reasons many of these utilities have been assigned a separate name and product number. Each separately numbered product that has a user interface also has separate documentation. For example, the KSAM intrinsics product contains code segments that are loaded into SL.PUB.SYS when the operating system is brought up, and these segments contain the procedures used by MPE V/E when it handles Keyed Sequential files. It has a separate product number and separate documentation[23], but it performs essential operating system functions.

All of the FOS components are included in the list of evaluated software, and those that require Privilege Mode are part of the Trusted Computing Base of the computer system. Privilege Mode is discussed on page 9, "Major Hardware Components." MPE V/E will allow users to call these privileged utilities, which execute with the Privilege Mode bit set before returning to User Mode after execution.

All the FOS utility programs, such as SPOOK5 and LOAD in MPE OPERATING SYSTEM and SECCONF in Security Monitor, are stored in group PUB.SYS. All intrinsics which are either callable by unprivileged users, or callable only by elements of MPE, are stored in SL.PUB.SYS and can be loaded as needed when a program is PREPed.

Most of the components of the TCB are written in Systems Programming Language (SPL), which is best described in *Systems Programming Language Textbook*[21], and most of them make use of calls to standard system subroutines, called intrinsics. For example, the GETPRIVMODE intrinsic allows a program that was invoked by a privileged user, and which was PREPed (PREP is the loader in MPE V/E) as a privilege mode program, to set its privilege bit. This effect is reversed with the GETUSERMODE intrinsic, thus allowing a program to run unprivileged except when it requires privilege mode. The scheduling software makes extensive use of intrinsics to manipulate the various scheduling queues. The exception to MPE components being written in SPL are the encryption utility in the Security Monitor product, and all of the Access Control Definition procedures within MPE Operating System. These are written in Pascal.

The elements of FOS which are part of the TCB are listed below, and each is described in a separate section following this list.

- MPE OPERATING SYSTEM (product HP32033)
- HP Security Monitor (product HP30392)
- ADCC (product HP32196)
- INSTALLER (product HP32433)
- KSAM INTRINSICS (product HP32208)
- MODCAL'LIB (product HP32047)
- NLS/3000 (product HP32414)
- TurboImage/3000 (product HP32215)

MPE OPERATING SYS (product HP32033)

The MPE V/E operating system kernel consists of a single product which implements all of the essential features that were provided on the original MPE operating system. This product contains all the intrinsics described in the *Intrinsics Reference Manual*[70].

These include access control checks, spooling of output, identification and authentication, the audit logging mechanism, and other functions. Other intrinsics and utility programs provided by a general purpose operating system, such as the scheduler, loader, memory manager, are also part of the product.

The central part of the operating system consists of several memory resident procedures, including the dispatcher procedure and the various memory manager procedures. Since these are memory resident procedures not associated with any process, they must have a data stack of their own. This is the Interrupt Control Stack which is also shared by the external interrupt handlers and non-trap type internal interrupt handlers, as described on page 18, "Domain Separation." These kernel procedures are interrupt driven. Any interrupt will cause the currently executing process or lower priority interrupt handler to stop executing, save its context on either the process stack or ICS, as appropriate, and execute the corresponding interrupt handler. Depending on what actions the interrupt handler has taken, it either exits from interrupt (IXIT command) or invokes the dispatcher (DISP command). The dispatcher determines whether to call memory management routines, which would occur, for example, if the interrupt was due to the currently running process calling a procedure in a code segment that was not loaded into main memory. Then the dispatcher determines whether the currently running process must be stopped and, if so, which process will run next. Since a clock interrupt occurs every 100 ms and no process may continue running past its third clock interrupt, the dispatcher will never go into a permanent wait state. See page 34, "Process Switching" for more information on process control.

HP Security Monitor (product HP30392)

In order to configure the system so it will run at the C2 level as described in the *Criteria*[86], the Security Monitor product is shipped with the system tapes. This product consists principally of a Security Configurator together with its support routines. The Security Configurator is run by the System Manager to enable all the features required to run in accordance with C2 requirements, and to enable other security features.

ADCC (product HP32196)

The Asynchronous Data Communication Controller, like the ATP (Advanced Terminal Processor) described on page 20, "I/O Manager", provides for terminal connections to the CPU. Although given a separate product number, this software is really part of the basic operating system since all terminals and serial printers that can be connected to the evaluated hardware (see page A-1, "Evaluated Hardware Components") use ADCC hardware and software. It requires use of privilege since it must access the lowest level device drivers and monitors in the system, which run in privileged mode. There are CPUs that run earlier versions of MPE but which do not support the ADCC hardware, so the supporting software was developed as a separate product

INSTALLER (product HP32433)

INSTALLER provides a software interface to customers when they are creating installation tapes that incorporate either newly acquired HP software, or else updates to existing HP software. Use of INSTALLER allows a system administrator to create a new load tape for a system that is otherwise identical to his existing system, except that the desired changes have been made. Since INSTALLER requires access to protected system files in order to copy their contents to the new tapes, it must run in privileged mode

KSAM INTRINSICS (product HP32208)

The Keyed Sequential Access Method is used by the KSAMUTIL utility program. (see page 25, "The MPE V/E File and Account System"). KSAMUTIL is used by the MPE V/E file system when it accesses KSAM files. This product is a collection of intrinsics that are used by KSAMUTIL. KSAM files are a part of the file system of MPE V/E but were introduced later than sequential files, and for administrative reasons the access utility and supporting intrinsics have separate product numbers.

MODCAL'LIB (product HP32047)

MODCAL'LIB is a single code segment required by programs written in the Modcal language (HP's Pascal plus extensions) to create and manipulate extra data segments.

NLS/3000 (product HP32414)

Native Language Support (NLS) consists of features which allow programmers to create applications which function correctly for users from a variety of countries and cultures using a variety of written languages. NLS/3000 users can use the GENCAT utility to create a catalog of messages, with one version of each message each in a different language, and store the catalog for use by an applications program. Then when a user of the applications program RUNs it, he can specify in which language he wants to see the messages produced by the application.

The NLS/3000 code is implemented using one or more extra data segments (XDS) per language. These segments are stored in the privileged group PUB.SYS, and MPE requires that a program be running in privileged mode to have any access, including read access, to data segments that are read into memory from a privileged group. Also, it is more efficient to run in split stack mode, in which the actual stack pointer is moved into the extra data segment before processing the data (see page 12, "Address Translation"). Also, several Asian languages require more information to be stored

about the language than will fit in the largest possible extra data segment, so when NLS/3000 processes messages in these languages it must use split stack mode. Split stack mode requires that the calling program be privileged.

TurboIMAGE/3000 (product HP32215)

TurboIMAGE/3000 is a data base management system which allows information to be logically related between data sets (files), minimizing data redundancy and providing fast information retrieval. All TurboIMAGE/3000 files are privileged files, so TurboIMAGE/3000 intrinsics require privileged mode. TurboIMAGE/3000 is shipped with all MPE systems.

Process Control

Each user is represented by his terminal session or batch job, and by other processes which can be created from this session, or which are created on the user's behalf by other system processes. In fact, most system processes that can be invoked by entering a command from the terminal session can also be invoked from within a user program by execution of a call to the COMMAND intrinsic [70]. This is termed programmatic access to an executing system process. A user with the proper capability can even create terminal sessions on other terminals using the STARTSESS intrinsic. However, all these processes are treated the same whether they are a Command Interpreter (CI) process, a batch process created by the STREAM command, or a system process which exists for the life of the system and is owned by user MANAGER.SYS. The security implications of process control are covered in the section on TCB Protected Resources (see page 37, "Subjects"). Details of the implementation for processes is given in the following section.

Process Manager

A process in MPE V/E can exist in one of four states: creation, activation, suspension, and termination. Once active, a process will remain active until it suspends itself, terminates, or is killed. Termination of a process terminates all of its descendents. Process address space is formed during the creation of a process. For more detailed information see page 37, "Creation and Destruction of Processes."

The stack contains the bulk of the process information in an area referred to as the Process Control Block Extension (PCBX). The PCBX is located in the stack starting at location 0 and extending to but not including DL. The PCBX is composed of three areas PXGLOBAL, PXFIXED, and PXFILE. The PXGLOBAL area of the stack contains job table indexes, job/session characteristics, and register data. The job/session characteristics include job type, job input and output logical devices, and an interactive or batch job flag. The register data includes DL and DB register values and an INIT Q location (relative to DB) of the first (oldest) stack marker. This marker is a marker for the

procedure TERMINATE because this prevents the user from doing too many exits. The PXFIXED area of the stack contains information for managing extra data segments. The PXFILE area of the stack is the file system's data base area and consists of Overhead, Control Block Table, Active File Table, and Available Space. A control block contains information on the file that is accessed and also, information on how the file should be accessed. Every file accessed by a process will have an entry in the active file table. This entry is used by the file system to characterize the file access and to give the location of its control blocks.

Process Switching

The dispatcher is responsible for saving the state of the current process when a swap occurs. This is done by calling the procedure SAVESTATE. When control is transferred from the user's stack to the ICS, the current value of the S register is saved on the ICS. SAVESTATE stores the value of S into the user stack's PCBX area. The remaining register values are already stored in the PCBX. After saving the state of the quiescent process, the dispatcher calls the procedure RESCHEDULE, which determines if any priority adjustments are necessary for this process. The dispatch then selects the highest priority process that is ready to run. If a process is ready to run and is not waiting for disc I/O or memory management activity, then the procedure LAUNCH is called to set the process up for execution. LAUNCH's primary job is to make sure that the stack and any current extra data segments are present in main memory. If not the process is flagged as absent and another process is selected for execution. If LAUNCH determines that the process is in fact able to execute, it must load the correct code segment table extension (CSTX) pointer into absolute location 1 of the ICS. The ICS is then set up to look as if an interrupt occurred in preparation to do an IXIT to actually transfer control to the process. IXIT is responsible for setting the code registers (PB, P, and PL). If the code segment is absent, a trap will occur and the dispatcher will be entered again.

Interprocess Communication

In MPE V/E interprocess communication (IPC) is a facility of the file system. IPC uses "message files" as the interface between processes. Message files act as queues of records. A process opening the file with read access is identified as a reader, and it may only read from the file. Similarly, a process opening the file with write access may only write to the file. If the same process needs to read and write to the file, it has to open the file twice; once as a reader and once as a writer. [13]

File access is coordinated by control blocks which contain information pertaining to a file. Control blocks are created at FOPEN time, consulted when a FREAD or FWRITE is done to the file, and pointers to the control block are deleted when a file is FCLOSED. A message file has three control blocks, each containing an IPC extension: a Logical Access Control Block (LACB), a Physical Access Control Block (PACB), and a File Control Block (FCB). Message files have multi-access capability which permits the sharing of access to a file. File access information which is unique and local to each individual accesser is contained in the LACB. File access information which is common

to each set of accessers is contained in the PACB. The PACB contains a FCB vector which points to the FCB.

The LACB extension contains accesser information specific to IPC and soft interrupts. The LACB is created at FOPEN time and is not shared with other processes. The LACB is found in the process's own stack in the PXFILE area.

The PACB contains the start of file pointer for the readers and the end of file for the writers. The PACB and extension is created by the first process to open the file. Since message files are always opened multi-access, the PACB and extension are located in an XDS. The PACB extension contains IPC specific information, including the location of reader and writer wait queues.

The IPC LACB and IPC PACB are used in combination to form the IPC Access Control Block (ACB). The ACB is a temporary structure that exists on the stack during FREAD and FWRITE. The ACB contains the most up-to-date description of the file for this accesser.

Tape and Serial Disc Processing

MPE V/E presents the same interface, as much as possible, to general users for each of two types of storage devices: serial and random access. The interface can exist at either the session level by issuing a system command or at the programmatic level by invoking an intrinsic. System discs containing the files within the group and account structure of MPE V/E must be kept unavailable to the general user, so the only interface that the general user has with these files is through the file structure described above (see page 25, "The MPE V/E File and Account System"). However, a user may directly interface with any of a number of serial devices. Of course, magnetic tapes are serial devices. But by issuing the command

>SERIAL logical_device

from within the VINIT subsystem, a privileged user can configure a normally randomly accessed device to act as a serial device. Disc drives so configured are referred to as serial discs. While serial discs are usually fixed within their drives, tapes can be mounted or dismounted by anyone who has physical access to the tape drive, including the user himself in a small facility where the users and system share the same physical control zone. The following discussion addresses the security constraints on both tape and serial disc media, but for the sake of brevity discussion will generally refer to tapes only. Identical procedures apply to serial discs unless noted otherwise.

Any serial medium can be either labelled or unlabelled. When the

LABEL=volume_number

parameter is specified in a FILE equation (a session command) or the FOPEN intrinsic (issued from a program) only a tape or serial disc that has an ANSI standard label with the correct 6 character volume number can be written; only a tape with an ANSI standard or IBM standard label can be read. Since standard labels includes the number of files, their names and sequential positions, any file can be accessed on a labelled tape; new files can be appended at the end of any file on a labelled tape, thus overwriting any files that were previously there. This is discussed in Chapter 9 of the *MPE File System Reference Manual* [25]. Any serial medium can be accessed by the commands FILE, STORE, RESTORE, and the intrinsics FREAD, FOPEN (which is called by the command FILE), FCLOSE, and FWRITE. Other intrinsics that can be used to access user data on a serial medium include FCONTROL, FREADBACKWARD, and FSPACE. The utility program FCOPY can also access serial media.

Mediation of access between users and the files stored on tapes occurs when any of the three commands FILE, STORE, and RESTORE are invoked by a user. However, if FOPEN is used to access a device file such as a tape drive with a user specified labelled or unlabelled tape on it, the intrinsics FREAD and FWRITE may be used to access any data on the tape without mediation by MPE V/E. If the tape is unlabelled, every byte on the tape is accessible; if the tape is labelled, the bytes of each separate file on the tape are accessible. Because of this, ACDs should be placed on all serial medium device classes, restricting access to them to privileged users. The commands discussed above and their usage are described in the *MPE/V Commands Reference Manual*[69] and in the *System Operation and Resource Management Reference Manual* [68]. Information on the intrinsics comes from *MPE V Intrinsics Reference Manual*[70].

Appendix D contains evaluators comments. The first section has a description of the STORE and RESTORE commands, and explains how the TCB provides mediation to the individual copies of files that are written to or read from serial media. (see page 83, "TCB Mediated Access to Serial Devices"). If access to serial media by unprivileged users could be restricted to these commands, then such users would not have to be restricted from accessing serial devices. But as currently implemented, access to these devices must be restricted to privileged users for secure operation of the computer system.

TCB Protected Resources

Subjects

MPE V/E has one type of subject - processes. These processes are divided into three subtypes: user processes, Command Interpreter (CI) processes and system processes. System processes service the entire operating system with functions such as device recognition, system auditing and spooling. They are all associated with the user MANAGER.SYS and are considered part of the operating system. Actions taken by system processes are distinguishable from actions taken by the user MANAGER.SYS[2]. A Command Interpreter process parses user commands and then calls the appropriate internal system procedure on the user's behalf, to carry out the commands. One CI process is associated with each user who logs on to the system. User processes are processes started by a user running programs.

Creation and Destruction of Processes

Processes are created in MPE V/E explicitly by a user, implicitly by a user, or by the system to perform a specific function. System processes are created at system initialization and execute with a higher priority than other processes. The normal lifetime of a system process is from system startup to system shutdown, though some are temporary such as those created to handle individual audit entries. CI processes are created by the system for every logon by a user, whether interactive or through a batch process created by the JOB command. A CI process lives until the user (or batch job) ends the session/job. When a user RUNs a program, a process is created and executes the proper code segments. If a user has process handling capability (PH), processes may be created from within program code using the CREATE or CREATEPROCESS intrinsics. The user indicates to the intrinsic a program file from which to create a process. After creation, the new process must be started by using the ACTIVATE intrinsic. The new process is a "child" of the process that creates it.

A process can delete itself using the TERMINATE intrinsic, or can delete any of its child processes with the KILL intrinsic. A user can delete processes by breaking out of execution using the BREAK key on the terminal and issuing the ABORT command. When a process is deleted, all of its descendants are deleted also. Order of deletion is as follows. Within a process tree structure, the newest generations are deleted first. Within each generation, processes are deleted in the order of their creation. Process termination causes all code and data segments in the process and all resources owned by the process to be released; all files opened by the process are closed; and finally, the Process Identification Number is released. [70]

Data Structures Involved in Process Creation

Whenever an action takes place in the system that causes a process to be created, a place in the Process Control Block (PCB) is allocated for it and a Process Identification Number (PIN) is assigned. The PCB entry contains state, control, and accounting information for each process that must always be in memory for use by the dispatcher. (e.g. stack pointers, priority.) The process control information that is not necessary to the dispatcher is kept in an area of the user stack called the PCBX (PCB eXtention). Next, the code segments for the process are loaded and appropriate table entries are allocated. Code segments in SLs (page 16) get entries in the CST and are numbered 0 - 277 (octal). Code segments from program files and relocatable libraries get entries in the CSTX (CST eXtension) and are numbered 300 - 377 (octal). The main difference between the CST and the CSTX is what parts are visible from each process. Each process sees the same CST, but has its own CSTX. So, CST entry 133 for processA and processB would always point to the same code segment while CSTX entry 302 would most likely point to different code segments depending on which process was referencing it. Finally, a data stack local to the process is set up and initialized and an entry is allocated in the Data Segment Table (DST). At this point, the system notifies the dispatcher that a new process is available to be scheduled. When a process is executing, the hardware is able to find all the pertinent process pointers and tables through an area in fixed low memory.

The HELLO command is used for interactive logon. It is issued at the colon prompt issued by the DEVREC (Device Recognition) system process. DEVREC uses identifying arguments specified in the HELLO command to initiate an interactive session by creating a CI process to act on behalf of a user. Arguments that may be specified on the HELLO command are session name, user name, user password, account name, account password, group name, group password, terminal designation, CPU time limit, process priority, and input priority. Unique identification is required on the HELLO command syntax so that the CI process created will represent a unique user. The CI process will prompt for any more information which is necessary for a session to begin (e.g. passwords) before allowing a user to issue further commands. Defaults for all the arguments except the passwords (and user name) may exist if the system manager has set them up for the user name. The session name argument can be used to differentiate between two sessions initiated by the same user in a single group and account. The Trusted Facility Manual (TFM) emphasizes that the system should be configured so that passwords are always required. For more information on the relation of user names in the system structure see page 25, "The MPE V/E File and Account System."

Certain roles and privileges may be associated with a subject at the discretion of the system manager. For more information on these capabilities see page 55, "Description of Privileges."

Objects

MPE V/E processes manipulate objects which include files, data segments, code segments, non-file interprocess communications (MAIL), job/session variables, main memory, the directory, registers, cache domains, and physical media such as serial disc volumes and tapes. [2]

Files in MPE V/E are subdivided into permanent disc files, device files, message files, spool files, and session-based files ($STDIN, $STDLST, $OLDPASS, $NEWPASS, temporary files). Data segments can be system tables, user defined information (extra data segments), or stacks. Job/session-based variables (JCWs) are objects shared among processes within a job/session. Main memory can be fixed low memory, region headers, or region trailers.

Named Objects

Named objects in MPE V/E have an acceptable Discretionary Access Control (DAC) mechanism. The objects that meet the current definition of named object are: permanent disc files and device files. The DAC mechanisms implemented in MPE V/E are described fully on page 41, "Discretionary Access Control."

Job/session-based files, data segments, JCWs, and MAIL data are all visible at the TCB interface, but are not sharable between users. Code segments and cache domains are not visible at the TCB interface. Main memory, the directory, and registers are all under the direct control of MPE V/E and are not shared between user processes. For these reasons, then, these objects are not named objects and do not require DAC.

Access to physical media, including serial devices, in the MPE V/E system operating at level C2 is a privileged access. A complete discussion of the handling of physical media is in the Software Architecture section (see page 35, "Tape and Serial Disc Processing").

At the C2 level, storage objects are primarily of concern for the object reuse requirements since there are no labeling requirements. For a complete discussion of storage objects in MPE V/E, see page 58, "Object Reuse."

This page intentionally left blank.

TCB Protection Mechanisms

Discretionary Access Control

MPE V/E has several mechanisms for controlling the access to files and devices: Access Control Definitions (ACDs), which, under the current system version, provide the C2 DAC mechanism; the Access Matrix, which, under previous system versions, provided the DAC mechanism, but which only provides access control to the granularity of a group user, any user allowed to access the group as the logon or home group; file lockwords, which work in conjunction with the Access Matrix and are basically passwords on files; and, the release and secure commands, which temporarily suspend and restore the access restrictions set up in the Access Matrix.

Description of the Mechanism

MPE V/E employs Access Control Definitions (ACDs) to provide discretionary access control for files and devices. Each file/device can optionally have an attached ACD. An ACD for a file/device is a data structure which contains a list of pairs of the access modes each user has to the file/device and user specifications. Modes can be any combination of R(read), W(write), L(lock), A(append), X(execute), NONE, and RACD (COPY or READ the ACD), separated by commas. Specifications are defined as user by username.accountname, wild cards by @.accountname or @.@, or as users and wild cards separated by commas. By combining modes and user specifications, ACDs, such as the following, are formed:

ACD = (R:JOHN.DOE; W,A,L:@.DOE, @.PAYROLL; R:@.@)

ACD = (NONE:JIM.DOE, @.ACCTING; R,W:@.@)

The ACD includes a list of users who can copy and read the ACD itself. When an ACD is being used, the Access Matrix is overridden and the ACD is the only discretionary mechanism used to determine who can access the file/device. [11]

ACDs are checked when a user attempts to access a file or acquire a device. For printers, the ACDs are also checked before printing to a device. The system function, FACCCHECK, evaluates ACDs when determining what types of access a user has to a file. The system function, DACCCHECK, evaluates ACDs when determining what types of access a user has to a device. FACCCHECK and DACCCHECK will check what types of access modes a user has by first checking if the user has System Manager capability, Account Manager capability, or is the CREATOR of the file/device.

Otherwise, it returns those access modes allowed to the user by the ACD associated with the file/device if there is such an ACD. The check to determine whether a file is privileged and the user has access to it is performed in FOPEN. If there is no ACD associated with the file/device, then it returns those access modes allowed to the user by the Access Matrix. [11]

To access a file protected by an Access Matrix, a user must have access to the appropriate account, group and file, and an ACD must not exist. File access is defined by five different access modes: read, write, execute, lock, and append. There is a sixth mode, save, which does not pertain to files, but is actually part of group security because files are saved within groups. In addition, a file lockword can be employed to further restrict access to a file. [3] Furthermore, files may be released and secured. By using the RELEASE command, it is possible for a user to temporarily suspend the security restrictions on any file, with the exception of privileged files, that is, files that require the user to have privelege mode access in order to access those files, created by that user. This allows the file to be accessed in any mode by any user; in other words, it offers unlimited access to the file. However, file lockword protection is not removed by the RELEASE command nor are the file security settings recorded in the system modified. The effects of the RELEASE command remain until the SECURE command is entered, which restores the original security provisions of the file, in the current or a later job/session.

When using the Access Matrix, users are grouped into categories such as Creator, Group User, Account User, etc. for accessing purposes. The possible user categories are any user (ANY), account librarian (AL), group librarian (GL) (for more information on AL and GL see page 55, "Description of Privileges"), creator (CR), group user (GU), and account member (AC). The Access Matrix makes use of the CR category at the file level only. Access to a file can be restricted to the Creator only (most restrictive), Group Librarian, Group Users, Account Librarian, Account Users or to any system user (least restrictive) by explicitly specifying, at the time of definition or redefinition of the file's Access Matrix, the user categories that have access. At system generation time, the Maximum Protection Feature can be set, using the Security Configurator, to provide Creator only access as the default for newly created files. However, under the Access Matrix no facility is provided to grant file access privilege to a certain user outside of the account other than opening up the access for the entire user community. [3]

With the Access Matrix, access to a file can be restricted at the account, group, and file level. The default file access restrictions at the three levels combine to result in overall default file access restrictions as follows. For any file in the group PUB of the SYS account the access permitted is (R,X:ANY; W:AL,GU). Save access to the group is granted to AL and GU. For any file in any other group in the SYS account the access permitted is (R,W,X:GU). Save access to group is granted to GU. For any file in the group PUB of any other account the access permitted is (R,X:AC; W:AL,GU). Save access to group is granted to AL and GU. Finally, for any file in any other group in any other account the access permitted is (R,W,X:GU). Save access to group is granted to GU. When the default security provisions are in force at all levels, the standard user, without any other user

attributes, has unlimited access (in all modes) to all files in the logon group and the home group, and READ and EXECUTE access (only) to all files in the PUB group of the individual's account, and in the SYS account's PUB group.

Manipulating ACDs

Functions exist that allow a user to manipulate ACDs. These functions provide the ability to create, copy, and delete an ACD. Other functions allow a user to list (display) and modify the contents of an ACD. A user can manipulate ACDs using the ALTSEC command or the programmatic interfaces provided by the system to manipulate ACDs. These programmatic interfaces are invoked by calling an intrinsic. [11]

Creating an ACD

The NEWACD parameter of the ALTSEC command is used to create an ACD and associate it to a specific file/device. For example, the following command associates an ACD with the file named PRIVATE.DOC:

```
:ALTSEC PRIVATE.DOC,FILENAME;NEWACD=(R:SAM.ACCTING)
```

To create an ACD for a file, and therefore be owner of the ACD, the user must be the CREATOR of the file, or Account Manager of the account where the file resides, or System Manager. To create an ACD for a device, the user must be System Manager.

Reading an ACD

Reading an ACD is performed when copying an ACD and when listing an ACD. Only the OWNER(s) of an ACD and those users to whom the OWNER(s) gives read ACD permission can read (and therefore copy and list) the ACD.

Copying an ACD

The COPYACD parameter of the ALTSEC command is used to copy an ACD from one file/device to another file/device. For example, the following command copies the ACD associated with the file named MYFILE to the file named YOURFILE:

```
:ALTSEC
YOURFILE,FILENAME;COPYACD=MYFILE,FILENAME
```

A user can copy an ACD from one file/device to another file/device only if that user is able to READ the ACD from the source file/device and is allowed to create an ACD for the destination file/device.

Listing an ACD

The LISTDIR5 utility and the LISTF command allow the user to list an ACD associated with a file/device. Only the Owner(s) of an ACD and those users authorized by the Owner of the ACD can list the ACD. The OWNER of an ACD authorizes who can read the ACD by including a list of users within the ACD who have RACD (read ACD) permission. Use of wildcards is allowed and used in the same manner as presently used with the LISTF command which lists descriptions of one or more permanent disc files.

Modifying an ACD

This function provides the user with the capability of modifying an existing ACD. The changes that can be made to an ACD are as follows: add a "modes:user specification" pair, using the ADDPAIR parameter of the ALTSEC command; replace the modes part of an existing pair, using the REPPAIR parameter of the ALTSEC command; and, delete a "modes:user specification" pair, using the DELPAIR parameter of the ALTSEC command. Only those user specifications : modes pairs that are being added/modified need to be written.

Deleting an ACD

The DELACD parameter of the ALTSEC command allows the user to delete an ACD associated with a file/device. The Owner of an ACD is the only one allowed to Delete the ACD. However, a special feature is provided so that System Manager can reset all ACDs in the system.

Description of Group Structure

Accounts are the unique partition within MPE V/E. Belonging to each account is a set of users and a set of groups, and every file and every volume set definition belongs to some group. The user possesses no files directly, but rather he accesses files, which he is linked to as the creator, belonging to groups. Further information on this subject is in the Software Architecture section (see page 25, "The MPE V/E File and Account System").

How Defaults Are Handled

There are two possible ways to handle default access control for newly created objects in MPE V/E. One is to enforce a restrictive default access control on newly created objects through the use of the Maximum Protection Feature. The other option is to require the user to explicitly specify the desired access controls on the object when he requests its creation. When an object is created and there is no ACD associated to the object, if the Maximum Protection Feature is set, the system will set the File Access Matrix so that only the creator of the object is granted access. The File Access Matrix wou d then appear as (R,W,X,L,A,S:CR). [11]

Access Revocation

Access to a file/device can be revoked by deleting the ACD entry of the user from the ACD associated with the file/device. The CREATOR of the ACD cannot be removed from the ACD nor can the CREATOR's access be restricted. The effect of the access revocation is delayed until the next attempt to access the file/device. If all entries of the ACD are removed, the ACD ceases to exist and DAC reverts to the Access Matrix that existed before the ACD was attached to the file/device. [11]

Limitations on Propagation of Access Rights

The owner of an ACD has all permissions to the ACD. The creator of the file to which the ACD is associated, a user with Account Manager capability, or a user with System Manager capability are each considered to be the owner of a file. System Manager is considered to be the owner of a device. The owner is the only one who can create, delete, modify, list and copy the ACD.

RACD is the only permission type allowed to be given to other users. The owner can give users RACD permission, thereby allowing users to READ and COPY the ACD. [11]

Audit of Security Relevant Events

Auditing within the MPE V/E system is carried out by the system logging facility. System logging records the use of certain resources by accounts, groups, and users. In addition, system logging describes system usage by creating a running log of actual events, correlated with the job or session that caused each event. Twenty three types of events, fourteen of which are security relevant, are currently logged. This section contains a discussion of MPE V/E's auditing facility and its security related features.

Log File Identification

Log file names always take the form "LOGxxxx.PUB.SYS", where xxxx is the log file number, ranging from 0000 to 9999, and LOGxxxx.PUB.SYS is a disc file. When a log file is closed and deactivated, MPE V/E creates a new log file with the name LOG####.PUB.SYS, with #### being incremented by one greater than xxxx. [7]

Record Fields

The record fields of an audit record for MPE V/E contain the major sections of header and log information. The header is divided into sections which include the record type, the record length, a timestamp, and a job/session (J/S) number. For each logon by a user, MPE V/E associates with it a unique job/session(J/S) number. The J/S number will be associated with the CI process that runs on the user's behalf. All auditable actions can be traced to the responsible user through the job/session number. [7]

Auditable Events

The events that are monitored by the system logging facility are recorded on log records contained in a disc file. Each event is recorded in one logical record. The events which are auditable fall into four categories: file system events, other object events, user control events, and administrative and privileged events. The logging of each of these events can be enabled by executing the SYSDUMP command, followed by the execution of a COLDLOAD [8].

File System Events

The following are auditable file system events: the creation of objects which includes files, directories, and other file objects; the deletion of objects which include files, directories and other file objects; modification of object access which includes access fields and ACDs; opening of objects which includes file open, mapping into process address space, and anything that causes an object to be manipulable by the process when it was not previously; the closing of objects which includes all operations that are the reverse of "opening"; and unsuccessful access to objects using the FOPEN command.

Other Object Events

The following are other auditable object events: operations on processes which include creation, deletion, etc.; removable media requests which include requests by users for removable media, as

distinct from the actual mounting and dismounting, which is performed by operators; removable media events which include the physical mounting and dismounting of labeled tapes and discs - actions performed by operators in response to user requests; I/O device use, which includes operations performed by users to get, release, and control non-file system I/O devices (but not I/O itself); and, other object operations which include operations on other objects, depending on the system.

User Control Events

The following are auditable user control events: logon and logoff, which includes all creation or destruction of user processes, or other requests made to that interface; unsuccessful logons which include unsuccessful logon or connection attempts - if a valid user ID is given, then this is logged; operator logon and logoff which consists of entry and exit of operators; and, user and group administration which includes creation and deletion of user and groups, and changes in group membership. These are usually administrative actions and require use of privilege. Such events are described in the next item.

Administrative and Privileged Events

The following are auditable administrative and privileged events: operator activities which include all operator actions and all operator input - operators will take some actions that can be detected by the hardware, but all operator typed input is recorded as well; privileged operations which include all activities of the administrator and all activities performed by a privileged user - since the administrator or operator can place the system (or his console) into some form of "privileged" mode, the facts of entry and exit are recorded, and, in any case, the privileged user may have the capability to disable further recording. In sum, operator, administrative, and privileged activities that occur during normal operation are recorded.

The only auditable events which are not optional are the "Logging Enabled" event, an I/O error, MPE V/E Maintenance Request, and Diagnostic Control Unit events.

Selectivity

LISTLOG5 in MPE V/E provides the ability to selectively review the audit records of the actions of one or more users based on individual identity. The possible selections, some of which make use of the wild card character ("@"), are as follows: a specific user in a specific account during a specific job/session specified by (job/session name, username.acctname); a specific user in a specific account during any job/session specified by (username.acctname) or (@,username.acctname); and, any user in a specific account during any job/session specified by (@.acctname) or (@,@.acctname). [71]

Assurances Against Loss of Audit Records

A feature exists that, when enabled, preserves audit logging integrity when a system logging error occurs. The feature, assurance of auditability option, is available as one of the global security options.

There are two types of system logging errors that can occur: recoverable errors, which result from managerial errors during the creation and management of the log file, and irrecoverable errors, which are either physical input/output errors or unit failures. When a recoverable error occurs, the system switches the console to the hard (physically connected) console, logs off all users except the hard console if the console user has OP or SM capability, sets the system in a single user mode, and sends a logging suspended error message to the console. After the error is corrected, a command, RESUMELOG, is required to resume the logging facility and the system will need to be configured to allow users to log on to the system. When an irrecoverable error occurs, the system takes the same course of action as for a recoverable except that a "logging stopped" error message is sent to the console. The system then requires a shutdown and restart. [11]

Who May Process The Audit Trail

A System Supervisor, who has OP capability, can configure the log files, create and close log files via the SWITCHLOG command, and display the status of the log file currently being used to record system events using the SHOWLOG command. [11]

The System Manager has the capability to purge log files. The System Manager also has the capability to run the program LISTLOG5 which analyzes system log files on MPE V/E. When given a log file name, LISTLOG5 then displays a numbered list of event types for which histories can be printed. LISTLOG5 then creates spool files of, or outputs to the terminal, the history of the events that were requested. [71]

Log File Security

Log files are created by, and therefore belong to, the system logging process. By implication, their creator is the original system manager, MANAGER.SYS. The current log file is opened exclusively (non-sharable) by the system logging process, thereby prohibiting any user from opening, and consequently modifying, the current log file, too. Once the log file has been closed, MPE V/E changes all the file access restrictions on the file from "ANY" to "CR" (creator) only. Thus, only MANAGER.SYS can access closed log files. [7]

Processing of Log Records

The records contained in the current system log file are available for processing after the OPERATOR executes the SWITCHLOG command which closes the current log file before it is full and creates and opens a new log file. The records in the current log file are not available until after the log file is closed because the current log file is opened exclusively.

Identification and Authentication

When a user logs on, the system attempts to authenticate the logon ID. The system checks its directory for the existence of the ID, then verifies the user's identity by checking the password (if required). In MPE V/E, passwords exist at three levels: user, group, and account. User passowrds prevent unauthorized persons from accessing the system. Account passwords protect the information in an account from users who are not members of the account. Group passwords protect files in the group from non-group members. If account and group passwords are created, their use is required for all users. Individual passwords may be set as optional or required. Account level passwords are created and maintained by System Managers. Group level passwords are created and maintained by Account Managers and System Managers. The System Manager can restrict an Account Manager from establishing groups with passwords by disabling the NEWGROUP and ALTGROUP commands and preventing any user, including Account Managers, from creating and altering groups. User passwords are created and made required by System and Account Managers, and can be changed by users.

How Users Are Added and Deleted

Users are identified to MPE V/E by users with System Manager or Account Manager capability using the NEWUSER command. The user name, user's password, the new user's capability list, subqueue name (i.e., the name of the highest priority subqueue that any job or session in the account can request for executing processes), the new user's local attribute and home group name are specified as parameters for the NEWUSER command. A capability list determines which commands the user may execute, whether the user can initiate sessions or jobs, save files, or use extra data

segments. More specifically, the capabilities that can be specified in a users capability list are as follows: System Manager, System Supervisor, Account Manager, Account Librarian, Batch Access, use communication software, diagnostic attribute, extra data segments, Group Librarian, interactive access, multiple RIN, Network Administrator, Node Manager, use nonsharable devices, use private disc volumes, privileged mode, process handling, programmatic sessions, save user files permanently, use user logging facilities, and create volume sets. The user's local attribute describes unique user capabilities that the user has for special applications. Users can be removed at two levels: as the member of a purged account or as a purged user. The command PURGEACCT removes an account and its groups and users from the system directory or from the specified volume set's directory, and use of this command requires System Manager capability. The command PURGEUSER removes a user from an account and requires Account Manager or System Manager capability. The user will not be affected until the next logon. An attempt to purge MANAGER.SYS will always fail, since this user can never be purged. If files created by a purged user remain after the user is purged from the system, the System Manager can remove them with the PURGE or PURGEACCT command, or the Account Manager or System Manager can eliminate them by executing PURGEGROUP.

Password Management

Authentication data or passwords are protected on MPE V/E in two ways. First, the passwords are stored in the Directory, a system table which can only be accessed in privileged mode. Second, the passwords are stored one-way encrypted using the encryption algorithm provided. [7]

Passwords on the USER, GROUP and ACCOUNT levels in t' directory are encrypted. In addition, device passwords are always encrypted. This is a one way encryption with the passwords being salted (passwords are concatenated with some other value) before encryption. Once passwords are encrypted, commands which display passwords (e.g., LISTUSER, LISTGROUP, and LISTACCT, all of which require Account Manager or System Manager capability) will not print out the unencrypted text for the passwords, but only show them in the encrypted form. The utility LISTDIR will not display a password in encrypted form; a token indicating that the password is encrypted is displayed instead.

Password text must begin with an alphabetic character, can contain up to eight alphanumeric characters, and can be entered in either upper or lower case. The text is upshifted prior to encryption, thus the letter case is not significant. [5]

The PASSWORD command allows users to establish or change their own user level passwords.

The System Manager can take a number of actions to control how the individual user manages his password. Some of these actions are taken by giving a command, or adding a particular parameter to a command. Other actions require the use of the Security Configuration. These actions are listed below:

Required Password Prompts on Interactive Logons

To prevent logon passwords from being seen by unauthorized people while they are being entered or afterwards by scrolling down the terminal, a system manager can enable the Password Prompt Required functionality by calling the Security Configurator. This means the operating system will reject any embedded password and always prompt for the needed ones with echo suppressed.

User Password Required Function

When creating a new user or altering an existing one, the account manager is able to specify whether or not that user is required to have a user password. Similarly, the system manager can specify whether all users in an account are required to have user level passwords when that account is created or altered. The user password requirement takes effect the next time a user tries to remove (by blanking out) the password. The user password requirement information is kept in the directory for the respective user or account.

Expired User Password

The expired user password capability allows the account manager to cause user passwords to be expired. This results in the user having to replace his old user password with a new one during his next logon. The user name will be invalid (i.e., the user will not be able to logon) until the password is updated. System managers are able to globally expire all user passwords and cause the expiration to happen automatically at certain time intervals. Global password expiration is only applicable to users who are required to have a password.

Minimum Length Password

System Manager can configure a systemwide value for the minimum password length. This will only affect users that subsequently alter their current password. If the minimum length is changed to a higher value, users who have passwords lengths of

less than the new value will not be affected until they try to change their current passwords. Minimum length is applicable to all password levels, that is, USER, GROUP and ACCOUNT.

Terminal Logon Password

The system manager is able to specify, on the logical device basis, the logon password for a specific device. If a terminal is configured as having a password, when a user attempts to logon the system will prompt for and verify the password before displaying the ":" prompt to let user input the HELLO, JOB, or DATA commands. Terminal passwords are kept in the security DST. Prompting for and verifying the device password is done in DEVREC which checks if the device requesting service has a password and acts accordingly.

Batch Submission Security

The STREAM command is used to spool batch jobs or data from a session or job and has optional time-related parameters associated with it that may be used to schedule jobs. Security exposure in batch submissions can be controlled through the Hewlett-Packard Security Monitor, using the Batch Submission Security Options of the Global Options Security Menu. Prevention of password exposure in batch submissions is effected by rejecting embedded passwords in job cards (the JOB command strings), prohibiting cross streaming (letting a user stream another user's job), and allowing System Manager and Account Manager to stream subordinates' jobs, and a user to stream one's own jobs without having to supply the passwords. A privileged interface, STREAMJOB, is provided which allows privileged mode programs to start jobs without having to supply passwords.

Embedded Password Disallowed

To prevent the exposure of passwords which may be embedded in job files, the security administrator can enable this option, which will cause MPE V/E to reject any !JOB command with password(s) embedded in it. When this feature is enabled, the STREAM command will not accept the optional embedded password syntax on the JOB card, regardless of the validity of the passwords. The feature is applicable to all JOB cards regardless of whether the job is from a disc file, from tapes or card readers, from within a job, or from interactive terminal input. The default for this feature is OFF, but the *Security Management Guide for MPE/V System Administration* recommends that, for maximum password protection, the feature be enabled.

Cross Stream Restriction

To preserve accountability of each individual user, the security administrator is able to disallow the cross streaming ability. This prevents a person without System Manager (SM) or Account Manager (AM) (of the appropriate account) capability from streaming jobs that log on as another person even if the first person knows the passwords of the job owner. When the feature is enabled, the following rules apply to batch streaming: a person is always able to stream his own jobs (jobs that logon to the same USERNAME.ACCTNAME as that person); AM can stream jobs that logon as any USER in the corresponding ACCOUNT; SM can stream jobs that logon to any USERNAME.ACCTNAME in the system; and, no other person can stream a job that logs on as a different user.

An exception to the last rule is provided, via the Cross Streaming Authorization option, to allow limited cross streaming on certain "protected" jobs, that is, a job that is streamed from a permanent file and logs on with the same ID as the creator of the file. A person, in addition to SM, AM and the job owner, is authorized to stream a protected job when he has EXECUTE access to the job file and the security administrator has enabled the Cross Streaming Authorization feature. The Cross Streaming Authorization for Protected Jobs is a supplemental option to the Cross Streaming Restriction feature. Therefore, authorization is needed only when the Restriction option is ON.

Stream Privilege for SM, AM and JOB OWNER

Stream privilege refers to the ability for the System Manager (SM) to stream all jobs in the system, an Account Manager (AM) to stream jobs within the account and users to stream one's own jobs, that is, a job that logs on to the same USER.ACCOUNT as the person that streams it, without having to supply passwords.

The Stream privilege can be extended to other authorized users when streaming protected jobs, via the Stream Privilege Authorization for Protected Jobs option. A person, other that SM, AM or the job owner, is granted this privilege (no password verification) when they have EXECUTE access to the job file, they are authorized to "cross stream" this job, and the Steam Privilege Authorization for Protected Jobs option is enabled. The *Security Management Guide*

for MPE/V System Administration recommends that the system be run with Stream Privilege and not Stream Privilege Authorization.

-STREAMJOB'

STREAMJOB is a privilege mode interface provided for PM (privileged) programs to stream jobs with no password requirements. This interface allows applications and subsystems to start jobs on behalf of various users without having to store passwords or look for them in the directory. By default, this interface is enabled. However, the security administrator can use the security configurator to disable the interface. By calling this interface (when it is enabled), the privilege mode application effectively obtains two authorizations: the cross streaming authorization, and the stream privilege for bypassing password verification. Thus, it is the responsibility of the application to verify the identity of the user running it, to make sure the application is used as intended and by the proper person. If disabled, this interface will function like a regular STREAM command.

It is important to note that although Stream security is separate from and complementary to file system security, it does not replace it. In any case, the user needs to successfully open a job file in order to stream it. Stream security is checked after the input file has been opened and read to obtain the JOB command.

Management of Groups

The account is the major unit in MPE V/E. Associated with each account is a unique file domain, a set of users who can access MPE V/E through this account, and a set of groups which partitions the account's accumulated resources and divides its file domain into private subdomains. [68]

When a user logs on to MPE V/E, three basic elements must be defined: an identifiable unit to which system resources are allocated and charged (account), a local set of disc files which the user may access (home group), and a name (user name) which identifies the user to the system as having access to the account and group [68]. The user is prompted for the account password, if it exists, and, if successful, is then prompted for the user's password, possibly followed by prompting for group password, if it exists. [6]

How Audited

When a user attempts to logon, whether successful or not, the event is audited [8]. However, the user is given three attempts at entering the correct password at each level (i.e., account, user, and group). The third unsuccessful attempt constitutes an invalid logon attempt. Each time an invalid password is typed, a message indicating the event appears at the System Console and the message is logged [6].

Hacker Frustration

A user is prompted for the account password, if it exists, first. If successful, the user is prompted for the user password, if it exists, possibly followed by group password prompting. A user is given three attempts at entering the correct password of each type. Exceeding the three invalid password attempts invalidates the log on and counts as an invalid logon attempt on the log on device. The system manager can configure the maximum invalid logon attempts allowed for the system. Once the invalid logon attempt count on a device exceeds the maximum, the device is "DOWNed" or taken off the system. The device must be "UPed" by an operator, or automatically after a set period of time, before MPE V/E will recognize the device. [6]

Minimum Assistance Login is a feature that is enabled by the System Manager with a call to the Security Configuration, and is implemented on a systemwide basis. When enabled, users will not be guided through the logon sequence as they currently are in MPE V/E. If any error is encountered in parsing the logon command, the message "* INVALID *" will be displayed at the user's terminal. The operator console, however, still receives the message specifying the reason for failure. Existing friendly messages will continue to be used when the feature is not explicitly enabled. [5]

Description of Privileges

Within MPE V/E there are several roles which a user can assume, primarily System Manager, Account Manager, System Supervisor, Account Librarian, Group Librarian, and Standard User. The following section does not contain a comprehensive list of MPE V/E privileges, but rather an outline of the main user roles within the system. The information on privileges contained in this section was obtained from *The System Operation and Resource Management Reference Manual*[68].

System Manager (SM)

System Manager capability grants the user the capability to manage the overall system and create the accounts, and groups and users within accounts, within the system. The first user with the System

Manager Attribute is designated on the system tape furnished with the HP 3000 Computer System. The System Manager, in turn, can designate other accounts having the same, or a subset of, capabilities.

The System Manager's functions are as follows: create new accounts; modify accounts, groups and users; delete accounts; configure, manage, and audit system security; list accounts, groups, and users for record purposes; list file attributes; obtain reports for all accounts; store and restore any or all files on the system; designate User Defined Commands (UDCs) for all system users; and, specify the expected System Startup State for each possible startup, using the System Startup State Configurator. The System Manager has unlimited file access to any file in the system, but can save files in his account only.

Account Manager (AM)

Account Manager capability allows the holder to manage all users and groups within the account. The first manager for each account is designated by the System Manager when the account is created. The Account Manager can, in turn, assign the Account Manager attribute to other users in the account.

Within the account, the Account Manager's functions are as follows: create new groups and users; modify groups and users; delete groups and users; insure the security of the account; list groups and users for record purposes; obtain reports for their account; list account files; store and restore account files (some files may also require SM, OP, or privileged mode (PM) capability); and, designate User Defined Commands (UDCs) for all account users.

System Supervisor (OP)

System Supervisor capability allows the user to have day to day external control of the system. It allows the user to manage scheduling subqueues, alter the system configuration, maintain the system and user logging facilities, and display various items of the system information. The System Supervisor attribute can be assigned by the System Manager.

The System Supervisor's functions are as follows: manage the system log file facility; exercise scheduling control over processes; permanently allocate/deallocate code in the virtual memory; obtain certain system reports and information; back up the operating system; modify the operating system parameters; save any or all files for archival purposes on magnetic tape or serial disc; and, manage the disc caching facility.

System Operator

Unlike the System Manager, who is assigned SM capability, or the System Supervisor, who is assigned OP capability, no "System Operator" mnemonic exists which entitles the Operator to execute a special subset of commands. Instead, the Operator's role and responsibilities are derived from his control of the Console, a privileged resource, since commands the Operator uses may only be entered at the Console or be specifically permitted to users at the Operator's discretion. Note that there is precisely one system console, and it is always hard configured as device L20.

The Operator typically performs the following functions: start the system with one of five load options; monitor the execution of jobs and sessions; control the use of peripheral devices; control the spooling of input and output files; distribute Operator capabilities among standard MPE V/E users with the ALLOW, ASSOCIATE, and JOBSECURITY commands; back up user and system files; shut down the system; and, open and close communication lines, and use data communications equipment. There is a special set of commands, called Console or Control-A commands, which are only executable at the device currently designated as the System Console. These commands, which are available at the Console without the operator being logged on, are as follows: ABORTJOB, which aborts a job, scheduled job, or session; LOGOFF, which aborts all jobs and sessions and sets the job/session execution limits to zero; LOGON, which re-establishes the job/session execution limits that existed prior to the LOGOFF command; SHUTDOWN, which shuts the system down in an orderly fashion; ABORTIO, which aborts pending I/O requests for a device; and, REPLY, which allows the operator to respond to device requests. All other Operator commands require the Operator to be logged on.

Account Librarian (AL)

Account Librarian capability gives a user special file access modes for file maintenance within the account. This attribute is assigned by users with the Account Manager attribute. Note that this capability is only meaningful if ACDs are not employed.

Group Librarian (GL)

Similar to the Account Librarian attribute, but limits a user's special file access modes to his home group. This attribute is assigned by users with the Account Manager attribute. Note that this capability is only meaningful if ACDs are not employed.

Standard User

All other users, not specifically assigned one or more of the above user attributes, fall into this class of Standard User by default.

Object Reuse

The following information on object reuse was obtained from the MPE/V Object Reuse Statement[4].

Files

There are five classifications of files in MPE V/E: permanent disc files, device files, message files, spool files, session-based files.

Permanent Disc Files

When storage space which has previously been used is assigned to a new file opened by a user, an end-of-file marker is set to the beginning of the file. The file system prevents the user from accessing any data beyond the end-of-file marker.

All physical I/O is carried out using blocks. The file system pads partially filled blocks with fill characters when writing to files. Thus, a user can only read what was written. No information residue can be obtained from permanent disc files.

Device Files

See page 60, "Physical Media."

Message Files

Message files are handled as permanent disc files and, therefore, object reuse is handled the same.

Spool Files

Spool files are stored internally in MPE V/E as unnamed disc files. Thus, the object reuse discussed for permanent disc files applies here as well.

Session-based Files

There are several classifications of session-based files, namely $STDIN, $STDLIST, $NEWPASS, $OLDPASS, and temporary files. These files are not known outside a CI process tree, that is, they are known (sharable) within the same job or session.

$STDIN and $STDLIST
: These are either device files, spool files, or disc files. The object reuse discussion for each of these types of files applies to these session-based files.

$NEWPASS, $OLDPASS, and temporary files
: These are all treated like permanent disc files, other than the fact that they are temporary and known only within a single job or session. The object reuse discussion for permanent disc files applies to these session-based files.

Data Segments

There are two types of data segments in MPE V/E: extra data segments and stacks.

Extra data segments
: Extra data segments are extra pieces of memory that are manipulable, but not sharable, and are used primarily when there is a need for space to manipulate tables, etc., and the user does not want to use up space on the process stack. When a user calls the intrinsic GETDSEG, which allocates an extra data segment to the user, before a segment is assigned, the content is overwritten with zeros in both main memory and virtual memory. When a privileged user requests extra data segments through the intrinsic GETDATASEGC, the segment is cleared before it is assigned. However, when a privileged user requests extra data segments through the intrinsic GETDATASEG, the segment is not cleared before it is assigned.

Stacks
: Stacks are data segments and are therefore cleared upon allocation.

: Data segments used as stacks have a certain structure. Architecturally supported registers keep track of a process's stack areas. As procedures are called, parameters and local variables are dynamically allocated on the stack and cut back after the procedure exits. In most cases, the old content on the stack is overwritten with new data as new procedures are called. Normally this does not pose any object reuse problems since the stack can only be referenced legally through specific hardware registers. However, two cases exist which may bring about object reuse

questions. First, uninitialized local variables may contain information left by previous procedures. Second, a user process, using assembly language instructions, can read residual information from previous procedure calls. In both cases the information that may be exposed is left by procedures which belong to a single process. All information obtained in the above ways is only related to a particular process initiated by a particular user. Therefore, no information is disclosed.

Intrinsics and command executors which run on the user's stack may manipulate security relevant information there. All the intrinsics and command executors which run on a user's stack do work on behalf of the users. Two occasions cause security relevant information to be placed on the stack. First, file lockwords may be brought onto the stack when the user tries to open a file. In this situation, the lockword is cleared before the procedure is exited. Note that with the use of ACDs, file lockwords are no longer security relevant. The other occasion exists when a user tries to start another job or session and passwords may be pushed onto the stack. This activity has been restricted by the new batch submission mechanisms (for more information on batch submission security see page 52, "Batch Submission Security") and by disabling the STARTSESS command (C2 configuration). The procedure STARTDEVICE, which verifies passwords, clears the password from the stack before exiting. All intrinsics and command executors which manipulate passwords or file lockwords clear their image in the stack before they exit.

Cache Domains

Cache domains, main memory buffers used in caching, are allocated by the memory manager and are not initially cleared. Object reuse of cache domains falls under the category of object reuse for disc files since a user can only effect changes in the cache domains through disc file I/O requests.

Physical Media

MPE V/E supports the following physical media: tapes, discs (removable, fixed), serial discs, floppy discs, terminals, printers and cards. For tapes, removable and fixed discs, serial discs, and floppy discs, users can use a degausser to clear the contents. Since removable and fixed discs are accessed by users through the file system as files, object reuse for them can also be handled with the file EOF marker.

Printing Facilities

MPE V/E has two ways in which to handle print requests. The first is referred to as hot printing. Hot printing devices are not spooled and once a user is allocated to the device, the user owns the device and an ACD may be put on the device. The other method of printing is through spooling. When a print request is made to a spooled device, a spool file is created and thereafter handled by SPOOK5.

SPOOK5

SPOOK5 is a utility program which allows a user to list, manipulate, and transfer spooled device files (spool files) created and maintained by MPE V/E. Any user can access this utility, but some functions are limited to users with Privileged Mode, System Manager, or Account Manager capabilities. [71] These functions are as follows:

- OUTPUT, which stores output spool files on tape or serial disc, is available for System Managers and System Supervisors only.

- INPUT, which restores output spool files from serial storage to the system, is available for System Managers and Supervisors only.

System Managers or System Supervisors can access any spool file. Account Managers can access any spool file in the logon account. Standard Users can access any spool file that they create.

This page intentionally left blank.

EVALUATION AS A C2 SYSTEM

Discretionary Access Control

Requirement

> The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

MPE V/E implements a Discretionary Access Control (DAC) mechanism between all subjects and all named objects. The enforcement mechanisms that meet this requirement are object attributes called Access Control Definitions (ACDs). The types of access allowed/restricted by ACDs are R(read), W(write), L(lock), A(append), X(execute), NONE, and RACD (copy or read the ACD). Users authorized to specify an ACD for an object include the CREATOR of the object as well as users with System Manager (SM) capability or Account Manager (AM) capability. In the case where an object is created without an ACD specified, access to the object is globally restricted so that only the creator of the object has access. For a more detailed description of the access control mechanisms in MPE V/E see page 41, "Discretionary Access Control."

Conclusion

MPE V/E satisfies the C2 Discretionary Access Control requirement.

Additional Requirement (B3)

CHANGE: The enforcement mechanism (*e.g., accss control lists*) shall allow users to specify and control sharing of those *objects*, and shall provide controls to limit propagation of access rights. These access controls shall be capable of *specifying, for each named object, a list of named individuals and a list of groups of named individuals with their respective modes of access to that object.*

ADD: Furthermore, for each such named object, it shall be possible to specify a list of named individuals and a list of groups of named individuals for which no access to the object is to be given.

The ACD mechanism in MPE V/E meets this additional requirement through the specification of username.accountname or @.accountname, and by the use of the NONE mode of access.

Conclusion

MPE V/E satisfies[1] the B3 Discretionary Access Control requirement.

Object Reuse

Requirement

> All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation, or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

Applicable Features

All information in storage objects released to the system is inaccessible to other subjects. The file system writes an end-of-file marker to newly allocated files and prevents the subject from accessing any data beyond that marker. The physical I/O to the files is done using blocks. Partially filled blocks are padded with fill characters so that a subject may not read beyond what it has written. Data segments are overwritten by zeros in both main memory and on disc when they are allocated. This includes data segments used as stacks by the system. Tapes and floppy discs must be degaussed before being reused. For a more detailed discussion of object reuse see page 58, "Object Reuse."

Conclusion

MPE V/E satisfies the C2 Object Reuse requirement.

1 Although MPE V/E satisfies this requirement at the B3 level, it does not satisfy the assurance requirements above its rated level.

Identification and Authentication

Requirement

> The TCB shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall use a protected mechanism (e.g., passwords) to authenticate the user's identity. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Applicable Features

MPE V/E requires users to logon before they can perform any operations on the system. The logon process is initiated by either the HELLO or JOB commands. Within these commands, a user specifies an ID in the form of username.acctname, a group within the account to logon to (default is the user's home group), and passwords, *if they exist*, for the account, user, and group (password not required for the home group). It is possible for the account manager and system manager to configure the password facility such that user passwords are required at the user level.

All users are required to identify themselves to the TCB before they can perform any operations on the system. A legal user on MPE V/E is identified by a combination of an up to eight character user identifier and an eight character account identifier (username.acctname). This ID uniquely identifies the user to MPE V/E. Upon logon by a user, MPE V/E associates a unique job/session number with a CI process that runs on behalf on the user. All auditable actions can be traced to the responsible user through the job/session number. The exception to this involves the capability to execute ctrl-A commands at the operator console without logon. In this case only, authentication of operators is achieved through procedural means. Otherwise, authentication of operators is achieved through their logon id, OPERATOR.SYS.

Authentication data (i.e., passwords) is protected by a vendor-supplied encryption utility[1] and is stored in a system table which is only accessible in privileged mode.

See page 49, "Identification and Authentication" for the full discussion of this subject.

Conclusion

MPE V/E satisfies the C2 Identification and Authentication requirement.

Audit

Requirement

> The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, actions taken by computer operators and system administrators and/or system security officers, and other security relevant events. For each recorded event, the audit record shall identify: date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity.

Applicable Features

MPE V/E auditing is carried out by the system logging facility. The audit trail is maintained in log files created and owned by the system logging process. Since the current log file is exclusively opened by the system logging process, it is not sharable and no user can open, and therefore modify, it. Once the log file has been closed, MPE V/E changes all the file access restrictions on the file

1 The encryption mechanism used was not evaluated by the evaluation team and no endorsement by the National Computer Security Center should be inferred.

from "ANY" to "CR" (the file creator) only. Therefore, only MANAGER.SYS can access closed log files.

Events that are recorded include user logon and logoff, unsuccessful logons, operator logon and logoff, creation of objects, deletion of objects, modification of object access, opening of objects, closing of objects, and unsuccessful access to objects. In addition, operations on processes, removable media requests, removable media events, I/O device use, and other object operations are also monitored and recorded. User and group administration, operator activities and privileged operations are events that are recorded as well.

The record fields of an audit record include the record type, noting the particular event which occurred, the length of the log record, a timestamp, and a job/session number which is associated with the CI process initiated for a user upon logon. For some security relevant events types, the success or failure of the event is recorded in a field called "error." For those audit record event types that deal with the creation and deletion of files, the file name is included in the record format.

The logon terminal ID of a user maps directly to an input logical device number. This input logical device number is recorded in the audit record for job initiation (i.e., logon) thereby providing traceability of the origin of a user logon.

LISTLOG5, MPE V/E's audit trail reduction tool, provides the ability to selectively audit the actions of one or more users based on individual identity. See page 45, "Audit of Security Relevant Events" for a complete description of this topic.

Conclusion

MPE V/E satisfies the C2 Audit requirement.

System Architecture

Requirement

> The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Applicable Features

The memory resident portions of the TCB execute in privileged mode and are thus protected from access by users other than privileged users during execution by the hardware virtual memory system. This includes the resident portions of MPE and the resident portions of the utilities that run in privilege mode.

Both the system utilities and operating system components are stored as executable files in group PUB.SYS. Loadable versions of intrinsics (system calls) are stored in the segment library SL.PUB.SYS. This group, like the entire SYS account, is owned by the designated user MANAGER.SYS. As long as the system administrator does not add any other users to this account, no user but the system administrator can tamper with these files. Appropriate warnings to this effect are in the Trusted Facility Manual. Therefore, the TCB maintains a domain for its own execution, and protects all its elements from external interference. See page 25, "Software Architecture" and page 25, "The MPE V/E File and Account System" for further information.

The TCB controls access to all resources to which a nonprivileged process has access, as described on page 37, "TCB Protected Resources." The TCB mediates all access between subjects and the system named objects through Discretionary Access Control mechanisms (see page 41, "Discretionary Access Control"). Access to other objects can be controlled by placing an ACD on them, making the devices subject to DAC. Otherwise, access is restricted to privileged users. All attempts at access are subject to audit (see page 45, "Audit of Security Relevant Events").

Conclusion

MPE V/E satisfies the C2 System Architecture requirement.

Additional Requirement (B1)

ADD: The TCB shall maintain process isolation through the provision of distinct address spaces under its control.

MPE V/E isolates each process within the individual code segments that make up that process. No instruction, either privileged or unprivileged, can modify the contents of a code segment of a running process.

Conclusion

MPE V/E satisfies[1] the B1 System Architecture requirement.

System Integrity

Requirement

> Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

MPE V/E provides hardware and firmware test packages which verify the correct functioning of hardware and microcode. For hardware, the offline Diagnostic Utility System (DUS) provides diagnostics to test disc drives, memory boards, terminal controller boards, and more. Additional diagnostics are provided through the Stand-Alone Diagnostic Utility Program and the Stand-Alone CPU Diagnostic which run without DUS. The SITT (System Integration Test Tape) package consists of COBOL, RPG programs, I/O programs, and system utilities which create a workload on a system in order to test the new firmware. CPU Selftest diagnostics, which are usually run by a diagnostic support processor (CMP,DCU), consist of FLDs (Fault Locating Diagnostics) which test the firmware.

A more detailed description of MPE V/E's diagnostics routines can be found in the following documents:

- HP3000 HP-IB Computer Systems Diagnostic Manual Set Volumes 1 and 2 [37]
- HP3000 Series 64/68/70 Diagnostic Manual Set Volumes 1 and 2 [75]
- HP3000 HP-IB Version Computer Systems CE Handbook [38]

1 Although MPE V/E satisfies this requirement at the B1 level, it does not satisfy the assurance requirements above its rated level.

- HP3000 Series 64/68/70 Computer Systems Customer Engineer Handbook [56]
- HP3000 Series MICRO 3000 Computer Systems Selftest Diagnostic [65]
- HP3000 Series MICRO 3000XE Computer Systems Selftest Diagnostic [63]

Some manuals describe general diagnostics:

- HP 3000 System Support Log [16]
- Stand Alone Sleuth Diagnostic [17]
- Stand Alone Diagnostic Utility Program II [18]
- Sleuth Simulator Diagnostic Reference Manual [33]
- AID Diagnostic Reference Manual [34]
- Diagnostic Utility System [35]
- Preface to Computer System Diagnostic Manual Set [36]

Systems:

- 64/68/70 DCU Self Test Diagnostic Manual [76]
- 64/68/70 IOMAP Diagnostic Reference Manual [77]
- 64/68/70 Stand Alone Diagnostic Utility Manual [78]
- 64/68/70 Stand-Alone CPU Diagnostic [79]
- 64/68/70 Memory Diagnostic Manual [82]
- 64/68/70 DMA Exerciser Diagnostic Reference Manual [83]
- Series 30/33 Maintenance Interface Diagnostic [43]

- Series 30/33 Coldload Self Test [41]
- Series 39/4X/5X CMP/System Selftest [54]
- Series 39/4X/5X IOMAP Diagnostic [48]
- HP3000 CS80 Device Diagnostic [80]
- Series 39/4X/5X DMA Exerciser Manual [66]
- Asynchronous Data Communications Channel Diagnostic Manual [44]
- General I/O Channel (GIC) Diagnostic Manual [46]
- Series 3X Memory Diagnostic [45]
- Series 4X Memory Diagnostic [55]
- Series 42XP/52/58 Memory Diagnostic [64]
- Stand Alone Memory Diagnostic [19]
- System Microprogram Verification Listing [24]
- Stand Alone CPU Diagnostic [26]
- Stand Alone Extended Precision Floating Point Diagnostic [27]
- Stand Alone Decimal Installation Diagnostic [28]
- Stand Alone Selector Channel Diagnostic [29]
- Stand-Alone Terminal Data Interface Diagnostic [30]
- Synchronous Single Line Control Stand Alone Diagnostic [31]

- Stand Alone Terminal Control Interface Diagnostic [32]
- Power Supply Diagnostic Set [60]
- Stand Alone System Clock Diagnostic [74]

Peripherals:

- 7902/9895 Flexible Disc Unit Diagnostic [47]
- Magnetic Tape Diagnostic Manual [39]
- 13037B Disc Controller Diagnostic [40]
- Disc Verifier Manual [42]
- Magnetic Tape Unit Diagnostic Loader [50]
- HP2680A/2688A Page Printer Verifier Diagnostic [51]
- HP7974A Magnetic Tape Drive Diagnostic [84]
- On Line HP Line Printer Verifier Diagnostic Manual [58]
- Option 333 Diagnostic Manual (Card Reader) [14]
- Printer Excerciser Reference Manual [49]
- Disc Memory Diagnostic Manual 7910K [52]
- Disc Error Rate Program Manual 7910 [53]
- On Line Diagnostic 2613A/17A/18A Line Printer [57]
- On Line Plotter Interface Diagnostic [59]
- Hardwired Serial Interface Stand Alone Diagnostic Program [62]
- Disc Driver Verifier Stand Alone Sleuth Program [72]

- CS80 Device Diagnostic Manual [81]
- CS80UTIL (See HP3000 CE Handbook) [38]
- TERMDSM (See HP3000 CE Handbook) [38]

Conclusion

MPE V/E satisfies the C2 System Integrity requirement.

Security Testing

Requirement

> The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. Testing shall be done to assure that there are no obvious ways for an unauthorized user to bypass or otherwise defeat the security protection mechanisms of the TCB. Testing shall also include a search for obvious flaws that would allow violation of resource isolation, or that would permit unauthorized access to the audit or authentication data.

Applicable Features

Functional Testing

During 6-9 September,1988, the format evaluation team tested the Hewlett Packard MPE V/E operating system running on the HP3000 model 70. Although the team encountered minor problems, the team is satisfied with the final outcome of the testing session. The full report of the details of the testing are in *Testing Final Report*[87], which was submitted to the Technical Review Board of the NCSC. The exact hardware configuration tested is listed on page A-3, "Test Configuration."

Team Comments

It was not obvious from the team members' study of the 36 hour long automated umbrella test suite, but it turns out that the way Hewlett Packard implemented these tests of the basic system functions (file open and close, login, spooling, I/O, scheduling, memory management, etc.) precludes their running with two security features that are in the set of C2 required features, as outlined in the TFM. The team determined that these features being either on or off would not affect the outcome of the functional tests, so these tests were run with all the other C2 features listed in the TFM. The two features that weren't turned on were extensively tested by their separate C2 umbrella test suite.

HP's own testing discovered a flaw in their new code, involving improperly logging the names of spool files when audit was turned on. This was corrected before the team arrived. The fact that they caught such an obscure error is one reason the team feels confident that the existing test suite is sufficient for functional testing.

The team found two additional flaws, one while looking at code while trying to figure out how to implement a team test, and one while running a team test. Such finds are expected from a C2 system, and the team members feel that this is evidence that a thorough job was done. Both errors were fixed during the test period and added to the patch tape. The entire automated test suite was rerun on the patched system.

The Software Testing Quick Reference Guide[10], which contains the script the test operator follows to set up the correct configuration for the automated testing, had some inexact entries in the script that led to the wrong configuration being set up. This caused lots of tests to abort or report an error. For example, the limit on number of jobs per session can be set, so the test suite tries values of 0, 5 and 7. Part of the configuration is the maximum number this limit can be set to, and is supposed to be 7 on the test system. But it was accidentally set to 5 due to the confusing script in the Quick Reference Guide. When the umbrella test set it to 7, it got an unexpected error message. There were other examples of this. Hewlett Packard has rewritten the Quick Reference Guide, and it will be available for future developers and quality assurance personnel.

Team Tests

The team members developed a suite of tests to cover some standard cases that might not be covered by the full set of functional tests developed by Hewlett Packard testing personnel. These tests and their resulting output are described on page D-1, "Description of Team Tests."

Conclusion

MPE V/E satisfies the C2 Security Testing requirement.

Security Features User's Guide

Requirement

> A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use, and how they interact with one another.

Applicable Features

The manual *MPE V/E Security and Account Structure User's Guide* [67], which is Hewlett Packard product number 32033-90136, has been evaluated by the team and found to contain the information specified by this requirement.

Conclusion

MPE V/E satisfies the C2 Security Features User's Guide requirement.

Trusted Facility Manual

Requirement

> A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given.

Applicable Features

The manual *Security Management Guide for MPE V/E System Administrators*[61], which is HP product number 30392-90001, has been evaluated by the evaluation team and found to contain the information specified for this requirement.

Conclusion

MPE V/E satisfies the C2 Trusted Facility Manual requirement.

Test Documentation

Requirement

> The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Applicable Features

Hewlett Packard has had a comprehensive testing program in place since well before MPE V/E was

first introduced. There is a separate Product Assurance team which interacts with the software development team through the design and implementation phases.

As described in the *Software Product Lifecycle*[15] manual, each phase in software production must be reviewed by Product Assurance personnel. At the end of the Design Phase, a test plan document must have been written by the development team and approved by Product Assurance. This Test Plan outlines the testing of the product that will occur during each of the design, implementation, user testing and release phases. Also during design phase, automated tests which will be used later are linked to the corresponding algorithms and structures, resulting in a complete review package. The automated test package is completed at the end of the implementation (coding) phase. It is augmented as necessary to reflect any unexpected problems that were found by test sites.

After testing at alpha and beta test sites, the development team assures that the final automated test package will run under the automated test umbrella.

In addition to *Software Product Lifecycle*[15], the other test documentation provided for MPE V/E includes the test plan and the actual automated test scripts that exercise the test cases. The test plan consists of two parts: an overview and a collection of test descriptions. The overview is in *System Tests B.01.08 CSY Software Quality Engineering Quick Reference Guide*[10]. This document shows how to set the system up for testing and lists the various tests.

The security monitor test plan, MPEAA Security, describes the specific tests for the access control system and other elements of the Security Monitor product that were added to the system previous to the evaluation. A separate C2 Security Umbrella test plan was devised for security functions added since the evaluation process began. Other test plan documents describe the tests for the file system, device access facility, spooling, logging (audit), and all other system functions provided by components of the TCB, which are listed in Appendix B (see page B-1).

Members of the evaluation team have inspected the following test plans and test scripts and found that they meet the requirements for test documentation:

MPEAA	Initial Security Features
MPEA	Store/Restore
MPEB	Operation Management
MPED	Accounting
MPEE	User Logging

MPEF	Spooling
MPEH	Process Handling
MPEI	RIN's
MPEJ	Native Language Support
MPEK	File System
MPEL	Device Tests
MPEM	Data Segmentation
MPEN	Miscellaneous Intrinsics
MPEO	JCW's
MPEP	Create Process
MPER	Expanded SL
MPES	Job Scheduling
MPET	Startup Configuration
MPEU	User Facility
MPEV	JCW's
MPEW	Session
MPEX	Private Volumes
MPEY	Auto Allocate
MPEZ	Utilities

C2 Security Umbrella
Remaining Security Features

Conclusion

MPE V/E satisfies the C2 Test Documentation requirement.

Design Documentation

Requirement

> Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Applicable Features

The MPE V/E philosophy of protection is described in the report *MPE/V Protection Philosophy*[1] with the implementation details of the associated operating system components further described in the following internal manuals:

- MPE/V Subjects and Objects [2]
- MPE/V Discretionary Access Control [3]
- MPE/V Object Reuse Statement [4]
- MPE/V Identification and Authentication [6]
- MPE/V Security Auditing [7], [8]

The TCB interface is described in the *HP 3000 Computer Systems: MPE V Commands Reference Manual*[69] for the interactive interface and in the *HP 3000 Computer Systems: MPE V Intrinsics Reference Manual*[70] for the procedural interface. Both manuals describe the syntax, parameters, and operation of each interface including condition codes and error messages.

The interface to utilities included within the TCB is described in the following manuals: *HP 3000 Computer Systems: KSAM 3000 Reference Manual*[23], *HP 3000 Computer System: NLS/3000 Reference Manual*[85], *HP 3000 Computer: INSTALLER Reference Manual*[22], and *HP 3000 Computer Systems: TurboIMAGE 3000 Reference Manual*[73].

The internals of the TCB are covered in the *HP Computer System Training Course: MPE Internals, Student Workbook*[12] which discusses in detail processes, scheduling, memory management, the file system, etc.

The principles of operation of the hardware and firmware are described in *HP 3000 Computer Systems: Machine Instruction Set Reference Manual*[20]. This manual lists all possible instruction formats. For each basic instruction interpreted directly by the hardware and for each element of the extended instruction set interpreted by the firmware, the resulting effect on all registers and memory elements is described. A description is given for all possible traps and interrupts that could occur as a result of executing each instruction.

Conclusion

MPE V/E satisfies the C2 Design Documentation requirement.

This page intentionally left blank.

EVALUATOR COMMENTS

Testing

The extensive quality assurance organization, and the manner in which testing is incorporated throughout the design, development, implementation and maintenance cycle for all Hewlett Packard software represents the best example of test documentation that any evaluator on the team has encountered in any evaluation. It is especially unusual to see testing addressed directly in the vendor's software engineering handbook[15]. In fact, it is fairly unusual to find that a vendor has a software engineering handbook for use by the development staff. These considerations greatly simplified the team testing of MPE V/E.

Configuration Management

Hewlett Packard has an extensive in house configuration accounting system that allows their development team to track all product code during development, testing and maintenance of their software. The system has evolved along with the MPE V/E system; not every software product uses this same system, but there is some kind of configuration accounting used on every product.

Hewlett Packard has expressed a willingness to also bring their documentation under configuration accounting, and to apply stricter configuration control to their development process. If this is done, then Hewlett Packard will have a configuration management system that is well suited to the Ratings Maintenance Program.

TCB Mediated Access to Serial Devices

MPE V/E provides TCB mediation of access to individual files stored on serial devices such as tapes and serial discs. In the revision currently under evaluation, there is no way to restrict an unprivileged user's access to the FREAD, FWRITE and FCOPY commands applied against serial devices, since these commands are also used to access disc files. Because of this, access to the serial devices themselves must be restricted to privileged users for secure operation of the computer system. However, if a future version of MPE V/E restricts access to tapes and serial discs to the use of the commands FILE, STORE and RESTORE, then general users should be allowed to access the information of these serial devices. The following discussion explains all possible ways an unprivileged user may access such objects using these three commands. The evaluation team believes that such restricted access would satisfy the requirement for DAC on named objects.

User Output Using STORE

An unprivileged user may only write out, or STORE, a copy of a disc file to which he has read access, and can only write to the serial medium, overwriting the previous contents, if any. Optionally, the user can issue a FILE command, such as

:FILE T;DEV=TAPE

to cause the name "T" to be associated with a tape. This is a formal association; the actual identification of the specific physical tape is a procedural matter. If FILE is not invoked, then the default is to associate the username with the tape. Then the command

:STORE file.descriptors;*T;[options]

is issued, where the file.descriptors are the files to be copied to the blank tape associated with the name "T." The asterisk means that T has been described by a previous file equation; otherwise, the command interpreter will not check to see if "T" has previously been defined to it. Optionally,

STORE file.descriptors;;[options]

will associate the name of the invoking user with the tape. The STORE command causes a request to appear on the operator's console to mount the appropriate tape on a free tape drive. The operator, or user if he has access to the operator's console and the tape drive, mounts the medium and responds to the message with the device number where the tape is mounted using the REPLY console command. At this point, the system checks whether there is an ACD on this serial device, and if so, checks whether the user is allowed access to the device. If access is allowed, then STORE copies the disc files, together with their complete specification as

filename.groupname.acctname

to the tape, the tape is rewound and dismounted. The user invoking the STORE command may optionally require the ACD be stored with the file on the tape. Any user who has read access to a disc file, either because the access matrix that allows someone in his group and account to read it, or because the user is listed on the ACD of a file with read access, may STORE a copy of that file to tape, thus becoming the creator of the tape file. He may keep the existing ACD or, as the new creator, leave it off. This is analogous to a user creating a disc copy of a disc file and becoming the creator of that file. The creator of the copy then has full rights to set the discretionary access to that copy.

There are two variations on this process, depending on whether the tape is to be labelled with an ANSI standard label.

Unlabelled Tapes

If the tape to be written need not have a label, then the FILE command is optional. The tape can be either delivered to the operator procedurally, or can be taken from a pool of unlabelled scratch tapes.

Labelled Tapes

If a labelled tape was given to the operator, or the tape is to be taken from a pool of labelled tapes, the FILE command must be used. In the former case

:FILE T;DEV=TAPE;LABEL=V47423

would cause the operator to mount the tape with that volume number on the label.

:FILE T;DEV=TAPE;LABEL=

would allow the operator to mount a labelled tape from a pool, then respond to a query from MPE V/E with the actual volume number from the existing label. In either case, the REPLY console entry is not required.

User Input Using RESTORE

An unprivileged user may only RESTORE a file from a tape or serial disc if he would be able to read the file if it were RESTOREd to the group and account from which it was STOREd. If the ACD of the file was stored with it on the tape or serial disc, the the ACD is checked. Otherwise, the default access matrix is checked. In either instance, use of the RESTORE command generates an audit log record.

There are several cases to consider:

Case 1 If a user in the same account/group as the files on the tape wishes to copy them back into the same group of the same account, he simply lists the file names. For example,

:RESTORE ;@

finds all files on the tape that have the same groupname and acctname as the user's. If the original files still exist, and the user has write access to a disc file with that name, they will be overwritten. If the user specifies KEEP as an option to RESTORE, disc files that already exist will not be overwritten.

Case 2 If the files on a tape belong to a group and/or account other than the one the user is logged onto, the user must have both read access to the files on the tape and write access to the destination group and account. To copy these files onto disc, the user issues the command

:RESTORE ;@;LOCAL;[other options]

The "@" character specifies all files and LOCAL specifies that the files are not to be copied into the filename.groupname.accntname that identifies them, but into the group and account that the user is presently logged onto. The filename is kept the same. This generates a message to the operator's console which is similar to the above, and the operator would find and identify the tape, mount it, and reply at the console with the drive number on which it was mounted. If the tape is labelled, the FILE equation with LABEL= option is required.

In this case, the user's access to the group and account of the files on the tape is checked. If they do not show read access, then the ACD of the file on the tape is checked. If no ACD is there, then the access matrix is checked. A general user would not have access to another group and account, but a user with SM, AM or OP privilege would be allowed access. The Commands manual [69] specifies that a "user must have read access to the files on the tape."

Case 3 If a user would be able to read a file if it were RESTOREd to a particular group on his same account, and the user has AM privilege (see page 57, "Description of Privileges"), then the GROUP= option would allow him to RESTORE the file to a different group than the one he is logged onto. Similarly, if the account on the tape file is different, but the user has SM or OP privilege, then he can use the ACCOUNT= option to copy the tape. Both of these require privileged capabilities, and are auditable.

Case 4 If a user is in a different account and group because the group or account of the files on the tape do not exist, he must have SM or OP privilege to RESTORE the files into an existing account. This is a use of privilege, and is auditable .

This page intentionally left blank.

EVALUATED HARDWARE COMPONENTS

MPE/V Hardware Configuration Range

System Processing Units

The MPE/V hardware configuration range includes the following System Processing Units (SPUs):

- MICRO 3000
- MICRO 3000XE
- Series 42
- Series 42XP/52
- Series 48
- Series 58
- Series 6x/70

Microcode

The MPE/V hardware configuration range includes the following microcode products, which must be loaded into the microstore of the processor that runs MPE/V:

- LOW END 3000 UCODE (product HP32515)
- MICROCODE (product HP32460)

Peripherals

The MPE/V hardware configuration range includes the following peripherals:

Disc Drives:

7933H, 7935H, 7936H, 7937H, 7933XP, 7935XP, 7936XP, 7937XP, 7957A, 7958A, 7914P, 7914CT, 7914ST, 7945A, 7920/7925M, 7920/7920S, 7914TD, 7911P/7912P, 7906M, 7906S

Tape Drives:

1/2 inch Magnetic Tapes:

7980A, 7978A/B, 7979A, 7974A, 7914ST, 7976A, 7970E Master/7914TD, 7970E Slave

1/4 inch Cartridge Tapes:

35401A, 9144A/7914CT, Integrated Tape

Printers:

Line Printers: 2567B, 2566A/B, 2565A, 2564B, 2563A/B, 26066A, 2608S, 2608A, 2611A/2613A/2617A/2619A

Page Printers:

2680A, 2688A

Serial Printers:

2686A/D RS-232, 2934A RS-232/422, 2933A RS-232/422, 2932A RS-232/422, 2564 RS-232/422, 2563A/B RS-232/422, 2687A RS-232, 2631B RS-232/422, 2603A RS-232/422, 2602A RS-232/422, 2601A RS-232/422, 33440A RS-232, 2684A RS-232

Plotters:

7220A/C/S/T, 7221A/B/C/S/T, 7225A/B, 7240A, 7245A/B, 7440A, 7475A, 7510A, 7550A, 7570A, 7580B, 7585A/B, 7586B, 7995A/7996A, 9872A/B/C/S/T

Other Devices:

37203A/37204A HP-IB Extender, 9895-010 Flexible Disc Drive, 30106A Card Reader, 26075A Multiple System Access Selector

Test Configuration

System Processing Unit:

Series 70, model 32471A with 8 MBytes of memory, 1 ATP and 1 HP2647F console device.

Disc Drives:

3 type 7935H disc drives installed as LDEV1, LDEV2 (private volume), and LDEV5.

Tape Drive:

1 7980A tape drive.

Printer:

1 26066A line printer.

Terminals:

2 type 2622A terminals, and 2 type 2627A terminals.

This page intentionally left blank.

EVALUATED SOFTWARE COMPONENTS

All files from the Master Installation Tape G.03.04, plus those on the patch number AV92, were installed on the system evaluated, and on the test system. This includes all elements of the TCB, and of Hewlett Packard Fundamental Operating System. The primary components installed are listed below.

TCB Components

- MPE OPERATING SYS (product HP32033)
- HP Security Monitor (product HP30392)
- ADCC (product HP32196)
- INSTALLER (product HP32433)
- KSAM INTRINSICS (product HP32208)
- MODCAL'LIB (product HP32047)
- NLS/3000 (product HP32414)
- TurboIMAGE/3000 (product HP32215)

Other FOS Software Installed on Test Machine

- COMPILER LIB (product HP32211)
- MITBLD GROUP
- BUILDINT/3000
- EDITOR
- FCOPY
- SORT/MERGE
- QUERY

- HPIB DIAGNOSTICS
- COBOL LIB
- MM/3000 MPE HOOKS
- ON LINE DIAGNOSTIC
- S 64 DIAGNOSTICS
- JAILAI
- CSD SUPPORT UTILITY
- COLOSSUS
- PCSERVER
- MIT BUILD TOOLS
- DUMMY DRVS/ACCT JO

REFERENCES

[1] MPE/V Protection Philosophy, July, 1988

[2] A. MPE/V Subjects and Objects, October 26, 1987

[3] B. MPE/V Discretionary Access Control Proposal, July 1987.

[4] C. MPE/V Object Reuse Statement, November 11, 1987.

[5] Commercial OS Security: External Specifications, June 4, 1987.

[6] D. MPE/V Identification and Authentication, October 27, 1987.

[7] F. MPE/V Security Auditing, July 1987.

[8] F. MPE/V Security Auditing (Addendum), November 11, 1987.

[9] HP 3000 Computer System General Information Manual, October 1984.

[10] B.01.08 CSY Software Testing Quick Reference Guide, September 1988.

[11] Phase II-A Security: External Specifications, November 20, 1987.

[12] HP Computer Systems Training Course: MPE INTERNALS Student Workbook.

[13] HP Computer Systems Training Course: APPENDIX/REFERENCE MATERIALS.

[14] Option 333 Diagnostic Manual (Card Reader), Part No. 02893-90005.

[15] Software Product Lifecycle, Part No. 5955-1756, February 1984.

[16] HP 3000 System Support Log, Part No. 03000-90116/7.

[17] Stand Alone Sleuth Diagnostic, Part No. 03000-90123.

[18] Stand Alone Diagnostic Utility Program II, Part No. 03000-90125.

[19] Stand Alone Memory Diagnostic, Part No. 30000-90004.

[20] HP 3000 Computer System: Machine Instruction Set Reference Manual, Part No. 30000-90022.

[21] Systems Programming Language Textbook, Part No. 30000-90025, September 1977.

[22] HP 3000 Software Update Manual, Part No. 32033-90036.

[23] HP 3000 Computer System: KSAM/3000 Reference Manual, Part No. 30000-90079, August 1986.

[24] System Microprogram Verification Listing, Part No. 30000-90136.

[25] MPE V File System Reference Manual, Part No. 30000-90236, February 1982.

[26] Stand Alone CPU Diagnostic, Part No. 30003-90001.

[27] Stand Alone Extended Precision Floating Point Diagnostic, Part No. 30011-90009.

[28] Stand Alone Decimal Installation Diagnostic, Part No. 30011-90010.

[29] Stand Alone Selector Channel Diagnostic, Part No. 30030-90011.

[30] Stand-Alone Terminal Data Interface Diagnostic, Part No. 30032-90011.

[31] Synchronous Single Line Control Stand Alone Diagnostic, Part No. 30055-90003.

[32] Stand Alone Terminal Control Interface Diagnostic, Part No. 30061-90004.

[33] Sleuth Simulator Diagnostic Reference Manual, Part No. 30070-90018.

[34] AID Diagnostic Reference Manual, Part No. 30070-90042.

[35] Diagnostic Utility System, Part No. 30070-90043.

[36] Preface to Computer System Diagnostic Manual Set, Part No. 30070-90044.

[37] HP3000 HP-IB Computer Systems Diagnostic Manual Set Volumes 1 and 2, Part No. 30070-60068.

[38] HP3000 HP-IB Version Computer Systems CE Handbook, Part No. 30070-90010.

[39] Magnetic Tape Diagnostic Manual, Part No. 30070-90015.

[40] 13037B Disc Controller Diagnostic, Part No. 30070-90016.

[41] Series 30/33 Coldload Self Test, Part No. 30070-90017.

[42] Disc Verifier Manual, Part No. 30070-90027.

[43] Series 30/33 Maintenance Interface Diagnostic, Part No. 30070-90028.

[44] Asynchronous Data Communications Channel Diagnostic Manual, Part No. 30070-90037.

[45] Series 3X Memory Diagnostic, Part No. 30070-90038.

[46] General I/O Channel (GIC) Diagnostic Manual, Part No. 30070-90039.

[47] 7902/9895 Flexible Disc Unit Diagnostic, Part No. 30070-90040.

[48] Series 39/4X/5X IOMAP Diagnostic, Part No. 30070-90041.

[49] Printer Exerciser Reference Manual, Part No. 30070-90045.

[50] Magnetic Tape Unit Diagnostic Loader, Part No. 30070-90073.

[51] HP2680A/2688A Page Printer Verifier Diagnostic, Part No. 30070-90074.

[52] Disc Memory Diagnostic Manual 7910K, Part No. 30080-90006.

[53] Disc Error Rate Program Manual 7910, Part No. 30080-90007.

[54] Series 39/4X/5X CMP/System Selftest, Part No. 30090-90005.

[55] Series 4X Memory Diagnostic, Part No. 30092-90001.

[56] HP3000 Series 64/68/70 Computer Systems Customer Engineer Handbook, Part No. 30140-90006.

[57] On Line Diagnostic 2613A/17A/18A Line Printer, Part No. 30209-90005.

[58] On Line HP Line Printer Verifier Diagnostic Manual, Part No. 30209-90007.

[59] On Line Plotter Interface Diagnostic, Part No. 30226-90006.

[60] Power Supply Diagnostic Set, Part No. 30310-90010.

[61] Security Management Guide for MPE V/E System Administrators, Part No. 30392-90001.

[62] Hardwired Serial Interface Stand Alone Diagnostic Program, Part No. 30360-90007.

[63] HP3000 Series MICRO 3000XE Computer Systems Selftest Diagnostics, Part No. 30474-90003.

[64] Series 42XP/52/58 Memory Diagnostic, Part No. 30477-90002.

[65] HP3000 Series MICRO 3000 Computer Systems Selftest Diagnostic, Part No. 30534-90001.

[66] Series 39/4X/5X DMA Exerciser Manual, Part No. 32003-90008.

[67] MPE V/E Security and Account Structure User's Guide, Part No. 32033-90136.

[68] MPE V System Operation and Resource Management Reference Manual, Part No. 32033-90005, February 1986.

[69] MPE V Commands Reference Manual, Part No. 32033-90006, February 1986.

[70] MPE V Intrinsics Reference Manual, Part No. 32033-90007, February 1986.

[71] MPE V Utilities Reference Manual, Part No. 32033-90008, February 1986.

[72] Disc Driver Verifier Stand Alone Sleuth Program, Part No. 32210-90001.

[73] HP 3000 Computer System: TurboIMAGE Reference Manual. Part No. 32215-90050.

[74] Stand Alone System Clock Diagnostic, Part No. 32230-90005.

[75] HP3000 Series 64/68/70 Diagnostic Manual Set Volumes 1 and 2, Part No. 32342-60001.

[76] 64/68/70 DCU Self Test Diagnosnc Manual, Part No. 32342-90002.

[77] 64/68/70 IOMAP Diagnostic Reference Manual, Part No. 32342-90010.

[78] 64/68/70 Stand Alone Diagnostic Utility Manual, Part No. 32342-90004.

[79] 64/68/70 Stand-Alone CPU Diagnostic, Part No. 32342-90005.

[80] HP3000 CS80 Device Diagnostic, Part No. 32342-90006.

[81] CS80 Device Diagnostic Manual, Part No. 32342-90006.

[82] 64/68/70 Memory Diagnostic Manual, Part No. 32342-90007.

[83] 64/68/70 DMA Exerciser Diagnostic Reference Manual, Part No. 32342-90008.

[84] HP7974A Magnetic Tape Drive Diagnostic, Part No. 32342-90011.

[85] HP 3000 Computer System: NLS/3000 Reference Manual. Part No. 32414-90001.

[86] Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, 26 December, 1985.

[87] Final Testing Report, Hewlett Packard Computer Systems Division MPE V/E, 4 October, 1988.

TEST DESCRIPTIONS

Description of Team Tests

The team members developed a suite of tests to cover some standard cases that might not be covered by the full set of functional tests developed by Hewlett Packard testing

AUDIT LOG Overflow

PURPOSE: to overflow the audit log space.

PROCEDURE: During the SYSDUMP that was run at the beginning of testing, logging of file open and close events was enabled; the Security Configurator (SECCONF) was run to turn on the assurance of auditability option.

A short SPL program was written that loops infinitely. It opens a file with a name defined in the program, appends the current time to the file, and closes the file each time through the loop.

Logon as MANAGER.SYS at the console. Obtain the current file space being used by PUB.SYS, the directory in which log files are contained. Set the file space limit for PUB.SYS to be a small amount more than the current file space being used.

Logon as an unprivileged user at a terminal connected to the test system and execute the SPL program.

Each time the SPL program loops it creates at least two log entries, one for opening the file and one for closing the file. By comparing the time values written to the file and those recorded in the log file, it is possible to determine any loss of log records.

RESULT: When the log file space filled, a logging suspended message appeared on the physical console device. All connected users were logged out. The test team used SM capability to inspect the file system and run LISTLOG5 on the latest log file. The log file was compared to the file written by the SPL program, and it was determined that no log records were lost. The ALTGROUP command was issued to remove the restriction on the space available to PUB.SYS and the RESUMELOG command was issued to continue auditing.

AUDIT LOG FILE Namespace Overflow

PURPOSE: to determine if the MPE system treats the unavailability of a unique next logfile name as a recoverable event.

PROCEDURE: Execute a SHOWLOG to determine the current log file (LOGxxxx). BUILD the next log file (LOG[xxxx+1]). Execute a SWITCHLOG to switch from the current log file to the next log file. Since the next log file already exists, a recoverable error should result.

RESULT: a logging suspended message appeared on the physical console device. All connected users were logged out. SM capability was used to inspect the file system, delete the next log file (LOG[xxxx+1]) which the team created, then issue RESUMELOG, which allowed the next log file to be opened for audit logging.

Execution of Undocumented Machine Operations

This tests each possible machine language instruction that is not fully documented for use on all machines listed in Appendix A.

PURPOSE: To determine that no security violations occur if an unprivileged user executes any undocumented machine language statement.

PROCEDURE: Upon investigation of the instruction set, it was found that all possible machine instructions are documented. A small subset, though, only have a meaningful definition on machines that are predecessors to the ones under evaluation. The instructions in that subset of undefined instructions were run on the test configuration. The test was also performed on the other hardware sets under evaluation. The results were the same on the 3x, 4x, and 7x machines.

An SPL program was created that forked a process to execute a supplied machine instruction. The undefined instructions were executed from the top of the stack using the XEQ 0 instruction. Instruction operands were pushed onto the stack previous to the instruction itself.

RESULT: Each instruction executed caused its process to abort with an error message indicating that an "undefined instruction" was attempted.

Execution of Privileged Commands by unprivileged User

When a program which contains instructions that require privilege are loaded by PREP, the resulting segment is marked to show that it must be run by a privileged user, or be run from within a privileged group. This is tested extensively in the umbrella test suite and appears to work properly. However, Hewlett Packard does not attempt to execute a privileged instruction from within a non-privileged code segment.

PURPOSE: This test is to assure the team that privileged instructions cannot be successfully executed by an unprivileged user.

PROCEDURE: The team went through the instruction set manually and identified all privileged instructions. The octal representations of the privileged instructions were place in a data file for use by the test program.

An SPL program was created that forked a process to execute a supplied machine instruction. The privileged instructions were executed from the top of the stack using the XEQ 0 instruction. Instruction operands were pushed onto the stack previous to the instruction itself.

RESULT: Each instruction executed caused its process to abort with an error message indicating that a "privilege instruction" was attempted. In the three cases where this result did not occur, it was determined that two of the instructions were not and did not need to be privileged. The documentation was in error. In the third case, it was found that the octal opcode identified in the documentation was in error. When the correct opcode was executed, it aborted as expected.

Protection of Process Privilege Bit

PURPOSE: To assure that a non-privileged user process cannot modify its status word while it is located on the user's process stack and thus gain privilege mode operation.

PROCEDURE: Develop a small SPL program that makes a procedure call so a stack marker was pushed on the process stack. The procedure that was called set the mode bit (bit 0) in the status word of the stack marker so it appeared the user was in privileged mode before the procedure call was made.

RESULT: Program terminated in an error state. Program error #6 PRIVILEGED INSTRUCTION. After investigation it was found that this was the proper error routine.

Protection of Address Space

PURPOSE: To assure a user cannot modify the CST number in the user's stack marker and exit out of a procedure call into a code segment belonging to another process.

PROCEDURE: Develop a small SPL program that makes a procedure call. The procedure that was called modifies the CST number located in the status word located in the stack marker on the user stack.

RESULT: Program terminated in an error state. Program error #30 - CST VIOLATION.

Object Reuse on Extra Data Segments

PURPOSE: To assure that the object reuse requirement is met on extra data segments. Extra data segments are to be cleared upon allocation.

PROCEDURE: Developed a small SPL program that requested an extra data segment. Upon allocation of the data segment the contents were moved to the user's stack and displayed to the screen. To assure the segment allocated was dirty before allocation the debugger was used to view the area on disk.

RESULT: The area allocated was dirty before allocation, and after allocation there was a clean extra data segment.

Ghost Interrupt Handlers

PURPOSE: To assure user mode processes cannot execute a ghost interrupt routine.

PROCEDURE: Developed a SPL program that attempted to execute the ghost routines located in ININ.

RESULTS: Program terminated in an error state. Program error #17 - STT UNCALLABLE.

DEBUG Attempted Security Violations

MPE V/E has a program called DEBUG which allows an unprivileged user to display or modify any word between DL and Z of his data segments, and display any word from PB to PL in any code segment of his currently loaded job. A privileged user can display or modify any word in any data segment of his currently running job, and display any instruction in any code segment in the current job.

PURPOSE: To use DEBUG on an unprivileged program to try to read or modify security relevant data, or to change to a privileged program.

PROCEDURE: A short program was written with a MAIN procedure that called a single procedure. The procedure puts an integer value on top of the stack, then executes a PCAL 0 instruction to execute the instruction on the top of stack. The program was RUN with the DEBUG option, and six different attempts were made to violate security.

- A copy of the status register of the MAIN procedure is kept in the base stack marker. This was changed from %060001 to %160001 (privilege bit set). DEBUG refused to make the change by ignoring the command.

- An attempt was made to display words between location 0 and DL in the data segment (this is the PCBX information). DEBUG issued a bounds violation message.

- An attempt was made to modify words between location 0 and DL in the data segment. DEBUG issued a bounds violation message.

- A breakpoint was set to interrupt the called procedure, its copy of the status register was modified by setting the privilege bit on, the procedure was allowed to continue but aborted due to a privilege violation.

- A breakpoint was set to interrupt the called procedure, then the return address on the stack marker was set to return to the privileged procedure COMMANDINTERP in segment library CIINIT (STT 2 in physical segment 44; start address 3141 in physical segment 44). The program was allowed to resume but aborted with a privilege violation.

- A breakpoint was set to interrupt the called procedure after the integer value was placed on the top of stack. This was replaced by a PCAL to the COMMANDINTERP privileged routine. When execution resumed the job aborted with a privilege violation.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION **UNCLASSIFIED**	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT **Approved for public release; Distribution Unlimited**
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-88/010	5. MONITORING ORGANIZATION REPORT NUMBER(S) **S231,332**
6a. NAME OF PERFORMING ORGANIZATION **National Computer Security Center** 6b. OFFICE SYMBOL (If applicable) **C12**	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code) **9800 Savage Road** **Ft. George G. Meade, MD 20755-6000**	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification) FINAL EVALUATION REPORT HEWLETT PACKARD COMPUTER SYSTEMS DIVISION MPE V/E	

12. PERSONAL AUTHOR(S)
Brown, R. Leonard; Donndelinger, James, J.; Jones, Jeffrey R.; Wilson, Anne M. (The Aerospace Corporation)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr, Mo., Day)	15. PAGE COUNT
Final	FROM TO	**881004**	**112**

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	**NCSC TCSEC DAC ACD OR HEWLETT PACKARD MPE V/E**

19. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Hewlett Packard Computer Systems Division's MPE V/E operating system has been evaluated by the National Computer Security Center (NCSC). The NCSC evaluation team has determined that MPE V/E satisfies all requirements of class C2 as specified in the DoD Trusted Computer System Evaluation Criteria (TCSEC) dated December 1985. MPE V/E also includes a number of features that enhance the security provided by the system. These features include both hardware features such as virtual memory and a two state architecture, and software features such as access control definitions (ACD) and individually restricted operating system commands.

This report documents the findings of the formal evaluation of MPE V/E.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED	21. ABSTRACT SECURITY CLASSIFICATION **UNCLASSIFIED**
22a. NAME OF RESPONSIBLE INDIVIDUAL **DENNIS E. SIRBAUGH**	22b. TELEPHONE NUMBER (Include Area Code) **(301)859-4458** 8b. OFFICE SYMBOL **C12**